In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

University of Alberta

Fragile Information Ecologies: How People Matter in Distributed Software Development Projects

by

Lorraine Kiel

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of *Master of Arts*

Department of *Anthropology* and
Department of *Computing Science*

Edmonton, Alberta
Fall 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisisitons et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-87576-8
Our file Notre référence
ISBN: 0-612-87576-8

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canadä

University of Alberta

Library Release Form

Name of Author: *Lorraine Ann Kiel*

Title of Thesis: *Fragile Information Ecologies: How People Matter in Distributed Software Development Projects*

Degree: *Master of Arts*

Year this Degree Granted: *2003*

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

September 24, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Fragile Information Ecologies: How People Matter in Distributed Software Development Projects* submitted by Lorraine Kiel in partial fulfillment of the requirements for the degree of Master of Arts.

Eric S. Higgs

H. James Hoover

Sean Gouglas

June 17, 2003

Abstract

The findings from a case study involving a mid-sized software development organization illustrate the complex interaction of factors common to many global software development projects. The focus of the study is a single product development group that was distributed between two international divisions of the company, one in Canada and one in Germany, for a twenty-month period. When the distribution ultimately failed, it was a web of social, cultural, linguistic and political factors, rather than use or misuse of specific tools or techniques, that emerged as being most significant in the project's ultimate demise. These factors are analyzed within a framework informed by technology studies, anthropology of work, and interdisciplinary distributed work studies.

Acknowledgements

First and foremost, my thanks go to the people at Chimera without whom I would have had nothing to research. Despite a hectic business environment and constant time pressures, the managers who sponsored my research and those who agreed to be interviewed gave generously of their time, their thoughts and their ideas.

I also thank my graduate advisors, Eric Higgs and Jim Hoover, who eagerly crossed interdisciplinary boundaries and somehow made it all work. Throughout my program, they were a constant source of encouragement and enthusiasm, and supplied considerable support for my research, both financially and intellectually.

The Departments of Anthropology and Computing Science also provided substantial financial assistance, along with wonderful administrators who frequently went out of their way to get things done, easing my and other students' passage through graduate school.

Finally, thank-you to the many friends and family who variously encouraged, challenged, supported and inspired me throughout my studies. Of special note are the Remedy gang, always willing to lend a critical eye and a good laugh, Jenaya Webb, a great friend who stuck it out with me in the playoffs, Trevor Tyre, who provided countless yummy snacks and took good care of Jenaya, and Mark Hughes, my reluctant editor, who keeps telling me that it's all possible.

Table of Contents

1. **Introduction** 1

2. **Theory** 5
   - 2.1. The Big Picture: Technology Studies 5
   - 2.1. A Closer Look At Two Approaches: Information ecologies and the device paradigm 8
   - 2.3. Anthropology of Work 16
   - 2.4. Distributed Work Studies 20

3. **Method** 31
   - 3.1. Setting up the Project 31
   - 3.2. The Interviews 35
   - 3.3. Data Analysis 39

4. **Observations** 41
   - 4.1. An Overview 41
   - 4.2. Themes 44
   - 4.3. Practices 64
   - 4.4. General Reflections on Distributed Development 80

5. **Analysis** 85
   - 5.1. Complexities of Communication 87
   - 5.2. Culture 94
   - 5.3. Practices that Amplify 97
   - 5.4. Ecological Integrity and Information as Commodity 99
   - 5.5. Concluding Remarks 100

6. **Conclusion** 103
   - 6.1. Some Guidelines 103
   - 6.2. On Future Research 109
   - 6.3. Infocentrism 112

7. **Bibliography** 114

8. **Appendix A - Sample Letters of Introduction** 119

9. **Appendix B - Sample Interview Consent Form** 121

10. **Appendix C - Sample Interview Questions** 122

11. **Appendix D - Ethics Statement and Approval** 123

List of Tables

Table 1: Practices 42

List of Figures

Figure 1: Biogeochemical cycle 85

1. Introduction

Despite a recent decline in the software development industry, there is every reason to believe that there will be continuing pressures toward the adoption of globalized approaches to software creation. These approaches may take the form of outsourcing agreements in which companies in Europe or North America contract out sub-sections of their software development work to businesses located in such regions as Asia, the Caribbean, or Russia. Often this is done because of sub-contractor specialization or because the cost of employing developers in these regions can be appreciably lower than in Europe or North America. Alternatively, global software development may involve collaborations among different divisions of large international corporations. Finally, smaller scale development efforts may consist of a small group of individual programmers who work together, in that they contribute to the same applications or systems, but who live in separate cities. In one form or another, globally distributed software development activities are likely to endure.

Consequently, managers will continue to be asked to organize software development projects that draw upon a mix of personnel in multiple locations. The technical barriers to such practices are diminishing rapidly. What about the human factors? Given the newness of the phenomenon, internationally distributed software development projects, whether they work, and why they either succeed or fail are little understood. The study presented in this thesis was undertaken in order to contribute to the small but growing body of empirical knowledge of global software development.

The subject company is a medium sized software development firm with operating divisions in Canada, the United States, Germany and Malaysia. While some of the divisions were created within the original corporate structure, others became part of the company through acquisition. Consequently, the company has a diverse and dispersed set of resources, and managing that diversity represents a significant challenge.

Given this international structure, it is not surprising that the organization has undertaken distributed projects in one form or another for much of its

history. Many of these projects have been short term, or structured on an ad hoc basis according to shifting delivery and development pressures. One project, however, represented a departure from this pattern.

For a period of roughly twenty months, spanning October 1999 to May 2001, the company structured all development of one of its core projects according to a fully distributed model. Before that time, they used multiple teams in their German and Canadian offices, with each creating customized versions of this core application for specific clients. Teams in Canada developed for North American clients, while teams in Germany handled clients in Europe. In order to reduce duplication of efforts, facilitate a comprehensive code re-use strategy, and address the problems of supporting and maintaining multiple, customized versions, they pooled expertise in the two offices. A single distributed team was created and given the task of producing a new base version of the core product.

Ultimately, the distribution failed, and the company consolidated all development for the base product in a single office. While a base version was successfully created, the significant overhead required to support the distribution was deemed to be too expensive. The purpose of my study was to answer the question, how did the distribution of software development for this product fail?

Three very distinct bodies of literature inform my research. The first is technology studies, which provides a framework for looking at technology in context, incorporating social and cultural factors along with the more technical considerations of how tools, taken in the broadest possible sense, do or do not function. The second relevant discipline, anthropology of work, contributes strategies for looking at what goes on in organizations and how people work. The third and final source of material is the rich, interdisciplinary body of research concerned specifically with distributed work. These distinct areas provide the necessary foundation to my research, and are explored in chapter 2.

In order to answer the question of how the distribution failed, I completed a qualitative case study in which I interviewed seventeen individuals representing a cross-section of people involved in the development process, including managers, business analysts, trainers, team leads and developers. I then applied

a systematic coding process in order to engage with the multiple perspectives of the participants, and to isolate recurring patterns in the interview data. A detailed description of my research method is presented in chapter 3.

Chapter 4 contains a comprehensive representation of my observations, organized into three main categories. The first consists of the five main themes of distributed development that emerged from the interviews: time, language, power, culture and trust. These themes illuminate the broad, conceptual challenges to distribution. The second category consists of the practices or specific ways of doing things on the ground, on a day-to-day basis within the company, as described by the participants. It is at the level of practices that the themes are realized and reinforced. Many of these practices involve specific tools such as e-mail, telephone, teleconference, configuration management, internet conferencing applications, translation tools, and the company intranet. Others refer more to business processes or ways of doing things, like software process, project team size, project scope, management practices and travel for face-to-face meetings. Finally, a third category contains more general comments that the participants provided about the process of distributed development. Together, the data presented in these three categories summarize the range of experiences with distributed development reported by study participants.

It is always compelling to try to isolate the one factor - the silver bullet (Brooks 1995) – that will solve a problem. In this case, the problem is how to engage the skills of geographically dispersed technical personnel to create software in an effective and economical manner. No single factor can be isolated as the cause of the failure of distribution described in this study. For example, while the practice of using e-mail has several characteristics that exist in any circumstance, certain properties become critical when employed in a predominantly asynchronous and politically charged communication environment. Other qualities emerge when messages are exchanged between people from different cultural or linguistic backgrounds. And still other factors are important when messages are exchanged between close friends. Examination of these contextual elements and their inherent interdependencies is fundamental to understanding how e-mail and other practices shape and are

shaped by the distributed software development project. This examination is the focus of chapter 5.

In chapter 6, I conclude by outlining general principles or guidelines for distributed development, things that may be useful to practitioners engaged in or thinking about entering into distributed software development arrangements. I then consider future areas of research suggested by my findings. Finally, I consider how my research challenges prevalent contemporary attitudes of *infocentrism* (Brown and Duguid 2000): the belief that everything can be reduced to information, and that more information is the solution to every problem.

2. Theory

The body of literature that informs this study and frames my theoretical approach consists of three main areas: technology studies, anthropology of work, and the specific research that deals with distributed work itself. The first two provide a disciplinary focus and explain the particular assumptions about what to study and how to study it. Technology studies contributes the broad theoretical context and establishes a strong basis for examining and understanding technology and technological systems. In the first two sections below, an overview of technology studies is presented, followed by a closer look at two particular approaches: information ecologies and the device paradigm.

The anthropology of work is concerned with understanding what goes on in organizations; a review of the relevant literature can be found in the third section of this chapter. It includes a consideration of the place of *culture* in studies of organizations, the impact of local and national cultures on the workplace, and what people do when they are at work.

The third source of material is the distributed development literature, a broadly interdisciplinary forum that spans such diverse fields as computer supported cooperative work, human-computer interaction, organizational theory, psychology, sociology and anthropology. An appreciation of how others have examined and come to understand distributed work in general, and distributed software development in particular, is a necessary complement to my own disciplinary approach to the topic. This is presented in the final section of this chapter.

2.1. The Big Picture: Technology Studies

One of the people I interviewed in the early stages of this project suggested that the ideal circumstances for undertaking software development was to have everyone involved in the process sit in a room together. He said that each step away from this ideal state introduces friction into the development process and

proposed that the key to successfully managing software development is to identify and minimize these instances of friction. In this study, I consider how a particular instance of distributed development technology - the tools and techniques adopted by the company – highlights possible sites of that friction. The field of technology studies provides a vital access point for this type of information: a critical reflection on how technology shapes and determines the form of what is a fundamentally human activity – software development.

What exactly do I mean by the *technology of distributed development*? I do not mean the workstations, file servers, fax machines, telephones, programming languages, compilers, revision control systems, e-mail systems, and computer networks that developers use on a daily basis. While these items are included in what technology is understood to be, they do not wholly comprise it. Instead, in keeping with the approach of technology studies, I employ an expanded definition of technology.

Pacey uses the term *technology-practice* to capture this expanded conception of technology: "the application of scientific and other knowledge to practical tasks by ordered systems that involve people and organizations, living things and machines" (Pacey 1983:6). Postman also argues against seeing technology simply as the tools that we use, and proposes that tools reflect a particular ideological bias, predisposing us "to construct the world as one thing rather than another, to value one thing over another, to amplify one sense or attitude more loudly than another" (Pacey 1993:13). He suggests that the particular tools we develop and use strongly influence how we think and what we think about. Franklin advocates that we consider technology to be a system that includes various ideas and practices and "entails far more than its individual material components. Technology involves organization, procedures, symbols...most of all, a mindset" (Franklin 1999:3). Pfaffenberger (1992) also uses the concept of a system, proposing that we not use the terms technology and tool synonymously, but instead refer to *sociotechnical systems*, a term that encompasses "the distinctive technological activity that stems from the linkage of techniques and material culture to the social coordination of labour" (Pfaffenberger 1992:497). He cautions against a simplistic approach that

assumes the tools employed in any given situation merely reflect the best way of completing necessary tasks (Pfaffenberger 1992). Instead, sociotechnical systems arise from complex social and cultural negotiations that determine what needs are, and how those needs are best addressed by the implementation of new technical solutions.

According to this expanded characterization of technology, understanding the technology of distributed software development requires that we consider not only the machines listed above (workstations, file servers, etc.), as well as the knowledge of how to use them, but also the ways in which people are organized to do their work, the administrative structures that are present, and the decision making processes that are followed. Also important are the broader social, historical and political settings in which the work is done - both organizationally and nationally - as well as the values, beliefs and ideas of the workers. These factors are part of technology studies, not peripheral to it. A comprehensive examination of technology embraces people as a part of the system, not as a frustrating complication to be integrated into analysis as an afterthought.

Ursula Franklin takes these factors into consideration when she discusses the overall impact that technologies have on our everyday lives. In her book *The Real World of Technology* (1999), Franklin posits that it is possible to become so immersed in the technological world around use that we become unable to see it's details. Her interest is in patterns of technological development, particularly information and communication technology. The pattern she describes is a shift from synchronous communication to asynchronous communication, a shift driven by dramatic changes in the power and availability of computer technologies. The problem is not so much that people communicate through asynchronous means. After all, since the invention of writing and later, of the printing press, asynchronous communication has been a prominent characteristic of many societies. Instead, Franklin notes that day-to-day, face-to-face communication is increasingly being supplanted by asynchronous approaches such as e-mail and voice mail. She argues that this is a problem, in part because an emphasis on information, on place-less and timeless approaches,

removes people from the biosphere and from a sense of time and place so critical to all past human civilizations.

2.2. A Closer Look at Two Approaches: Information Ecologies and the Device Paradigm

Within this general critical technology studies framework, two approaches in particular form a significant part of the analysis in this work: information ecologies and the device paradigm.

2.2.1. Information Ecologies

The first of these, proposed by Bonnie Nardi and Vicki O'day (1999), is concerned with the metaphors that we employ when we talk and think about technology. Nardi and O'day describe three metaphors that are often used to characterize technology: technology as tool, technology as text, and technology as system. They propose a fourth metaphor as a useful alternative to these common approaches: technology as information ecology. All four metaphors are informative in that they highlight particular aspects of technologies.

When we think about technologies as tools, we are drawn to considerations of what tasks the tool can be used to complete, or the "device-used-by-an-individual-to-get-something-done" (Nardi and O'day 1999:28). On the one hand, this metaphor is useful in that it highlights how we use technologies to do things, where designers have been successful at creating things for particular tasks, and, in particular, where there has been a failure in design leading to a poor task-technology fit. Nardi and O'day argue that when we think of technology as tool, we can attend to the tool's affordances, or the "set of all possible actions" (Norman 1999:122) available to us when we use the tool. This allows us to understand the subtle (or not so subtle) influences of the technology's designers. They suggest that for designers, the technology as tool metaphor can promote sensitivity to the fact that their creation will ultimately be used by a person: "it is not enough to think about the tool's inherent elegance

and capabilities; one also has to think about the handles it offers to users" (Nardi and O'day 1999:30).

However, there are limitations to the tool metaphor. While it is good for considerations of functionality, it is less useful when the discussion extends to broader questions of how the technology is located in a particular web of values and relationships. It is also less useful in helping us to understand the unanticipated ways in which tools can be used. This leads to the second metaphor: technology as text.

Drawing on the work of scholars such as Bruno Latour, Nardi and O'day contend that technology can be thought about as "a form of communication, a carrier of meaning that may be reinterpreted as the technology passes through different social situations (Nardi and O'day 1999:31)." Books, newspaper articles and other documents are both written and read. Their meaning is not fixed once the writer commits words to the page. Rather, meaning for both writer and reader is socially and culturally influenced; the writer's intended meaning can differ markedly from the reader's construction of meaning. Nardi and O'day suggest that the analysis of the writer-text-reader relationship is appropriate for studying the designer-technology-user relationship. We can start to *read* technologies, looking for what they communicate and the contexts in which they are written. When we think of technology as text, the roles that technologies play, and the motivations and goals of their designers become the focus: their functionality becomes less central.

The third metaphor that Nardi and O'day examine is technology as system. Systems thinking engages with social, political and economic contexts at the broadest level, including "the pervasive influence of technology on our lives" (Nardi and O'day 1999:33). One of the common characteristics of work that frames technology as a system is a pessimistic insistence that technology has considerable momentum, and that it has permeated our lives so completely that there are few options for any of us. There is a strong sense of technological determinism in the work of proponents of this view, the list of which includes Jacques Ellul and Langdon Winner.

Ellul argues for the existence of *technique*, which Nardi and O'day interpret as "a cultural mindset in which pure, unadulterated efficiency is the dominant human value" (Nardi and O'day 1999:34). They suggest that while his depiction of *technique* as an unstoppable, autonomous force may seem greatly exaggerated and over the top, at some level it resonates with our own experiences. For example, the current debate over human cloning seems to be circumscribed by a prevailing sentiment of inevitability. Researchers who study cloning are part of a larger system of science that seems to be permeated by the notion that what can be done, should be done. Nardi and O'day suggest that this illustrates the mindset that Ellul was writing about.

Langdon Winner's approach, while less pessimistic than Ellul's, reiterates the argument that the technical infrastructures that we assemble around ourselves have enduring impact and considerable momentum. Winner (1986) argues against the notion of the neutrality of tools. He acknowledges the argument made by social scientists that tools are embedded in social, political and economic contexts, and that these contexts shape what technologies we employ and how we structure them. However, Winner refuses to concede that studies of technology are indistinguishable from studies of other social, political or economic endeavours. Instead, he insists that technologies have political qualities in and of themselves: "there are...good reasons to believe that technology is politically significant in its own right, good reasons that the standard models of social science only go so far in accounting for what is most interesting and troubling about the subject" (Winner 1986:21).

Winner's theory of "technological politics" brings to our attention the sociotechnical systems in which we are immersed and explains how these systems condition the choices that we can make. As an example, consider the case of the highway overpasses of Long Island, New York (Winner 1986). Robert Moses was a prominent figure in the New York area for many years, and was responsible for many infrastructure projects undertaken between 1920 and 1970. During a time of significant highway construction, Moses stipulated that the overpasses had to be built with a clearance that ultimately restricted roadway access. Public buses, with a height of twelve feet, could not fit under the

overpasses, and were consequently unable to travel along the parkways; only private automobiles were granted access. Winner contends that this was an intentional move by Moses to determine who could get to Long Island: white, upper and middle class, car-driving individuals could; those who travelled by bus – often poor and African American - could not.

According to Winner, this example of a technology that had profound social effects illustrates how technologies as systems have considerable momentum. In some cases, the resulting social effects are explicit and intentional, and in many more they are implicit and often unanticipated. For all technologies, Winner argues, we make choices, especially in the early stages of adoption. Once those choices are made, and considerable investment in material, labour and facilities is complete, the systems become entrenched and further choices are highly constrained.

Ellul and Winner's work, along with others who make similar arguments (Postman 1992, Franklin 1999), shows us that thinking about technology as a system allows us to attend to some of the big-picture issues. The system, tool and text metaphors each supply powerful and diverse ways of understanding and talking about technology. Nardi and O'day propose a fourth metaphor, one that will be particularly useful for this work: technology as information ecology.

> We define an information ecology to be a system of people, practices, values, and technologies in a particular local environment. In information ecologies, the spotlight is not on technology, but on human activities that are served by technology. (Nardi and O'day 1999:49)

The ecology metaphor foregrounds certain things not captured by any of the other three: it allows us to consider the network of relationships in which technologies exist while acknowledging the particularities of locality. It enables us to consider such diverse characteristics as complexity, interdependency, diversity, change, and stability. They present a hospital intensive care unit (ICU) as an example of an information ecology:

> It has an impressive collection of people and technologies, all focused on the activity of treating critically ill patients. Human experts (nurses,

> physicians, therapists, ethicists) and machines (monitors, probes, and the many other devices in the ICU) all have roles to play in ensuring smooth, round-the-clock care. Though this is a setting with an obvious reliance on advanced technologies, it is clear that human expertise, judgment, empathy, cooperation, and values are central in making the system work. (Nardi and O'day 1999:49)

An ICU has diversity in the machines and individuals operating within it, complexity in the various tasks and roles, and interdependency among machines and human actors. The techniques they apply, the diseases they face, and the machines they employ vary considerably over time. Nonetheless, an ICU is relatively stable in that it is capable of adapting to change. Our attention is drawn to these characteristics when we invoke the ecology metaphor.

Nardi and O'day introduce the concept of a keystone species, which in a biological context describes a species critical to the very survival of the ecosystem. They suggest that the keystone species in information ecology terms is the person or people whose skills enable the technology. This may be the highly technical people who design and implement the system. On the other hand, it might be the expert user who facilitates system use on a day-to-day basis. For example, in a library – a clear example of an information ecology, according to Nardi and O'day – the librarians, not the systems designers or technicians, represent the keystone species. They argue that in many cases where new technologies have failed, it is because the role of the keystone species has been ignored or overlooked.[1]

A more extensive elaboration of the information ecology metaphor will appear later in this work. For now, this brief overview serves as an introduction to one of the underlying ways of thinking about distributed software development that I will be incorporating into my analysis.

---

[1] In a class on distributed software development that I was recently involved in, I came face-to-face with this phenomenon. The course made use of a video-conferencing facility that did not live up to the expectations of one of the instructors. There were frequent occasions where things failed and the system had to be reset by the attending technician. I saw the technician as playing a valuable role in the successful functioning of the facility. The instructor looked forward to the day in the near future when the technology would finally work *properly* and we wouldn't need the technician anymore. At our location, the university supplied a technician for the duration of our class; at the other location, one technician was split among multiple facilities, ignoring the role of a keystone species.

2.2.2. The Device Paradigm

Another useful approach is the *device paradigm*, as formulated by Albert Borgmann. Like Franklin, Winner, and other critics of technology, Borgmann notes that it is often a challenge to take a step back from contemporary life and think critically about technology. He suggests that technology is a "deeply ingrained pattern that...may be difficult or impossible to see" (Borgmann 1984:35). This ingrained pattern becomes refigured as a kind of common sense, "the obvious way of doing things which requires no discussion and, more important, is not accessible to discussion" (Borgman 1984:35).

Borgmann explicates the device paradigm by considering examples from recent history. Consider, for example, the woodstove. In the early part of the 20th century, the woodstove was the means by which most North American dwellings were heated. Keeping the woodstove operating was a matter of considerable human attention. The wood had to be cut, collected and stacked. Someone had to attend to the weather conditions, as a particularly harsh winter would necessitate the collection of more fuel. Each day, someone would have to clean out the stove and light a new fire, and throughout the day, the fire would need tending. Finally, as the main source of warmth in the home, the woodstove would be a common place of social gathering, not to mention the focal point of meal preparation.

While the production of heat was the central function of the woodstove, it served numerous peripheral purposes, and required significant engagement from those who used it.

Borgmann contrasts the woodstove with the central forced-air furnace. It is usually tucked away somewhere, perhaps in a basement or a closet. Because it regulates the internal temperature of the home through an automatic feedback control system, very little daily interaction with the furnace is necessary. Of course, the furnace is tied into a complex infrastructure of fuel supply, engineering and manufacturing. Nonetheless, these properties fade into the background. It is heat, and heat alone that the central heating system brings to

mind. Nobody needs to directly attend to it. People don't gather around it. It is not a focal point of the home.

Borgmann suggests the device paradigm as a way of talking about these two very different systems that appear to serve the same basic function: to heat the home. Both the woodstove and the forced air furnace provide heat, what Borgmann calls a *commodity*. But what differentiates the modern heating system from the woodstove is that heat is much more available from the former than from the latter. It is easy to turn up the thermostat in order to get more heat; it is not as easy to go outside, grab more logs, put them in the stove, and wait for them to burn and produce more heat that will eventually creep through parts of the house. Another striking difference between the two is the different level of physical engagement with the world that they require.

Finally, Borgmann suggests that in the movement from woodstove to central heating, the commodity –heat – becomes separated from the means through which the heat is generated – the machinery. He restates this as a separation of means and ends. In the case of the woodstove, the commodity and the machinery are tightly coupled, both in space and in idea. However, today it is easy not to even think about the machinery – the furnace itself – while enjoying the readily available heat.[2]

In Borgmann's terms, the woodstove represents a *focal thing*, while the forced air furnace is an example of a *device*. As a focal thing, the woodstove is much more than a heat provider; it is also a focus of many social activities and relationships (people cut the wood, people light the fire, people gather around the source of the warmth). The focal thing is "inseparable from it's context...its world, and from our commerce with the thing and its world, namely, engagement" (Borgmann 1984:41). While it does provide a commodity – namely, heat - the woodstove does so much more. This multiple purpose and the demand for physical engagement are characteristics of focal things.

---

[2] Of course machinery and commodity can be jarringly reunited, as in the case of events such as the ice storm that hit Montreal in January of 1998.

Devices, on the other hand, make commodities readily available to us: heat, for example, is "rendered instantaneous, ubiquitous, safe and easy" (Borgmann 1984:41). This is accomplished when we transfer all of the requisite commodity-producing activities into the machinery of the device. Consequently, the foreground of the commodity is sharply separated from the machinery of the device. The machinery fades into the background and is largely concealed form our attention.

How the commodity is produced becomes somewhat unimportant. We can pull out one chunk of machinery, replace it with a newer one, and voila, we have improved commodity availability. Thus devices are fungible and anonymous; whereas focal things are rooted in place and a network of social relationships that Borgmann calls *focal practices*.

Borgmann maintains that contemporary technological life is characterized by the primacy of devices over things. This pattern of separation through which focal things and focal practices are replaced by the machinery and commodity of the device is the device paradigm. He also invokes a new way of thinking about information.

According to Borgmann (1999), there are three types of information: information *about* reality, information *for* reality, and information *as* reality. The first consists of maps, charts and books – things that tell us about the world we live in and help us find our way through it. The second is comprised of instructions or techniques by which we can modify or create reality. For example, a recipe and a musical score are examples of information *for* reality. The third describes a type of information that is fundamentally technological in character, and is the type that Borgmann is most concerned about. His example is the compact disc on which, for example, a symphonic performance of a particular classical work has been recorded. The bits recorded on the disc, when scanned by the compact disc reading mechanism, processed, and ultimately converted to sound, are taken as reality and the original performance is lost. This can be restated in terms of the device paradigm: the focal practices centred on skilful and physically engaged musicians disappear, replaced by the commodity of music. Borgmann insists that we need to maintain a balance among the

different types of information, and that contemporary life is threatened by the predominance of information as reality.

How can the device paradigm and the relations between information and reality be employed in the analysis of distributed software development? In the same way that Nardi and O'day's application of the information ecology metaphor to technology highlights certain things, Borgmann's device paradigm and his typology of information draw our attention to particular facets of distributed software development. My main approach in later chapters is to talk about the foregrounding of *information* as a commodity, and the way in which the challenges of distribution are often reduced to problems of information exchange. I argue that the act of producing software is not merely a routine task of information production, but is instead a complex, ambiguous and socially embedded practice. When we begin to think of the conference call merely as an opportunity for information exchange, we miss the ways in which successful collaboration is, indeed, a focal practice.

2.3. Anthropology of Work

While the overarching perspective of this study is technologically oriented, a second point of view has significantly informed my work: the anthropology of work. There are two directions within the anthropology of work that I will consider here. The first is concerned with studies of organizations and organizational culture, and the second is concerned with detailed studies of particular work practices.

Van Maanen and Barley (1985) were among the first to establish the legitimacy of using the term *culture* in the context of the organization.[3] They suggest that the term *organizational culture* is particularly useful because it directs our attention to phenomena within organizations that are not addressed by other areas of organizational theory.

---

[3] The *organization*, in this case, includes small and large businesses, non-profit groups, governments, and public bodies such as schools and police forces.

> Culture implies that human behavior is partially prescribed by a collectively created and sustained way of life that cannot be personality based because it is shared by diverse individuals. Neither can a way of life be derived solely from structure, since members of separate collectives themselves occupy equivalent positions in a structural matrix. Rather, culture points to an analysis mediating between deterministic and volunteeristic models of behavior in organizations. (Van Maanen and Barley 1985:32)

Culture is a group-level phenomenon that is enacted and reproduced at the individual level through the actions and interactions that take place in a particular organizational context (Brannen 1998, Van Maanen and Barley 1985). Van Maanen and Barley suggest that we think of culture as "a set of solutions devised by a group of people to meet specific problems posed by the situations they face in common" (1985:33).

From its emergence, the organizational culture concept has been a widely adopted[4] and powerful tool in analyzing the behaviour of people in business settings. Much of the work relevant to this study investigates the point at which national cultures and organizational cultures intersect.

Mary Yoko Brannen (1998) considered this intersection when she completed a long-term project involving an American manufacturing facility that was purchased by a Japanese corporation. Brannen and others (Gluesing 1998, Reeves-Ellington 1995, 1997, 1998) note that most literature on organizational culture treats culture as some fixed body of knowledge that must be imparted to newcomers. She argues that this ignores the important role of individuals in enacting and negotiating culture. As culture is a dynamic and complex phenomenon that changes because of the interactions of individuals,

---

[4] It is not completely without critics. Writing at the same time that Van Maanen and Barley solidified the concept of organizational culture, Smircich (1985) argues that organizational theorists miss the richness of culture and what could be learned about organizations through cultural analysis. Instead, they focus on culture as a monolithic property of an organization, one that can be manipulated for organizational effectiveness. Smircich challenges researchers to really look at the life within organizations rather than thinking about some "intact structural package" (57) that is the organization's culture. While this is an important distinction, it is peripherally relevant to this project. Because of the limitations of my fieldwork, any discussions of *culture* in this work will be limited to what the participants identify as culture.

understanding changes in organizational cultures requires an investigation of what individuals in organizations do on a daily basis. Concluding from her research that the cultural negotiations that occur in international firms is complex and poorly understood, Brannen advocates for in-depth case studies and the micro-level analysis that ethnographic research affords to the researcher concerned with the dynamics of organizational cultures.

In her study of global teaming, anthropologist Julia Gluesing (1998) examines the process that unfolds when employees from different offices in a multi-national high technology manufacturing firm, involving diverse cultural and organizational environments, work together on a single project in a manner that crosses traditional boundaries. She identifies five strategies of global teaming: (1) communicating, (2) integrating relationships, (3) structuring process, (4) doing work, and (5) facilitating. Gluesing also specifies two underlying factors essential to successful global teaming: balancing power and building connections. Like other researchers, Gluesing concludes that organizational culture is neither stable nor monolithic. In addition, members of organizations bring any number of diverse cultural identities into the workplace where they are continuously maintained, enhanced and negotiated.

What ties much of this work together is that the subject is the organization, or particular sub-groups (or sub-cultures) within the organization. Another informative perspective within anthropology considers the activities of work itself, a highly particularized local phenomenon.

One of the most widely cited examples of this perspective is the ethnography of photocopier repair technicians completed by Julian Orr (1996). Orr immersed himself in the work of several repair technicians employed by Xerox in southern California. He began his project by attending an internal repair personnel course, and then tagged along to repair calls, service meetings, and other activities comprising the workdays of several technicians. In doing so, he revealed much about the multitude of tasks comprising what he described as the highly skilled work of technicians. Orr describes the minutiae of work as the *practice*: the *doing* parts of the job. He characterizes this practice in the case of the repair technicians as being highly skilled, complex, improvisational, and

highly resistant to rationalization or decisive codification. On the one hand, Orr describes, technicians were expected to follow the procedures manuals in order to complete their repairs. This manual was the source of "official" causes for failures. However, it was seldom the only resource relied upon by the technicians. In fact, actual work practices differed substantially from the methods defined in machine documentation.

Technicians constructed an elaborate set of *stories* to describe particular problems, customers, and machines, and shared these stories, creating a rich body of knowledge that all local service personnel participated in.

Orr argues that while work is central to many aspects of contemporary life – identity, self-worth, economic success, organizational effectiveness – detailed accounts of what people actually do when they work are rare.

> Authors may presume that modern jobs are in some sense known...This black box treatment of modern occupations denies that there is anything interesting or problematic about the work itself; surely one goal of industrial anthropology must be to open these black boxes and see if the goings-on within are those we expect. (Orr 1996:153)

He maintains that the complexities of specific work practices are invisible from a distance, yet managers and other policy makers operate from that distant position and consequently have a poor vantage point from which to make informed decisions. Orr's arguments resonate with many others who do workplace anthropology (eg. Baba 1998, Suchman 1995, 1999, Ngwenyama 1998).

Brown and Duguid (2000) turn their gaze to the field of business process reengineering using a similar line of reasoning. They contend that activities typically directed toward radically changing – reengineering - the workplace privilege *process* over *practice*. Process describes the systematic way in which work is expected to be completed: the rules about how things are done; the routines followed in completing tasks. Processes are codified, formalized and written down. Individuals are not prominent in discussions of process. Instead,

groups of tasks that need to be completed are more likely to be addressed. Workers, then, are more like plug-compatible work units.

Practice, on the other hand, is "the activity involved in getting work done" (Brown and Duguid 2000:97). Practice involves *local knowledge*, networks of people, and conversations that may or may not bear directly on the task at hand. The practice of the workplace is reflected in and by shared work experience. When we want to understand practice, we need to know the workarounds that people rely on when official procedures are inadequate. Practice is individual, local and varied: it is messy, concerned with understanding the nitty-gritty of how to get things done, not just what is to be done.

Brown and Duguid argue that process and practice need to be seen as alternative perspectives on the workplace. Both are important: neither can be disregarded. If you're going to do a good job of process redesign, you must understand and account for practice.

The work of these researchers exemplifies the mantra of workplace anthropology: if you want to understand the work, you need to go to the people who do the work – preferably while they are doing it. My goal in this project is to undo the screws on the black box of distributed development and peek inside.

2.4. Distributed Work Studies

The final topic to be considered here is the body of literature concerned specifically with distributed work. The first area of interest consists of the practitioners' guides. Written by people with significant experience at actually doing distributed work, they present excellent overviews of the various reasons organizations might choose to undertake distributed work arrangements, highlight the many pitfalls, and generally suggest practical steps for managers and workers to try. For example, *Virtual Project Management*, by Paul E. McMahon includes such sections as "Culture, Adversity, and Communication", "Collocation at the Right Time", "The Team Loyalty Tale", and "How to Effectively Use Virtual Communication Technologies" (McMahon 2001:xvi – xix). This book is pragmatically organized to allow the distributed project manager to

get an overview of the problems she is likely to face, identify probable causes for these problems, and pinpoint promising solutions. It is long on prescription for successful projects, but short on data and analysis.

Another popular practitioner's guide is Erran Carmel's *Global Software Teams: Collaborating Across Borders and Time Zones* (1999). A collection of personal anecdotes, research findings, and extensive personal experience, Carmel presents a comprehensive guide to various forms of global software development projects. He outlines things to consider regarding coordination and control, suggests ways to effectively divide projects across globally dispersed teams, and proposes the strengths and weaknesses of various communication strategies. Carmel also discusses the impact of such factors as time zone and cultural differences.

While these largely prescriptive works may be useful sourcebooks for those engaged in organizing distributed projects, they are less useful for understanding the specific processes underway in particular work situations. Fortunately, distributed work has become an active area of interest. What follows is a brief introduction to a substantial body of useful and relevant scholarly literature on the topic.

One of the more obvious areas of inquiry is concerned with software outsourcing arrangements. Nicholson and Sahay (2001) adopt an explicitly political framework in their study of the global software outsourcing arrangement between a British software development firm and an Indian outsourcing company. They argue that corporations "eager to reap the flexibility and economies offered by developing offshore" (Nicholson and Sahay 2001:26) often do not understand the associated cultural, social and political complexities. Nor are the objectives of offshore development necessarily made explicit to all parties involved in the process. The researchers found that there was much more underway in their case study than the simple maximizing of economic factors. Management of the British firm used the Indian outsourcing exercise to dramatically transform operations in their home office. They wanted to establish "a coherent organisational culture" (Nicholson and Sahay 2001:33) within the corporation to offset the diversity of approaches and work styles that was a

remnant of the recent spate of corporate acquisitions. Software outsourcing was one strategy in the overall homogenization process. The Indian programmers brought structured techniques that were disseminated throughout the company in part because these techniques were seen to improve development but also because they significantly improved control over how developers worked. Management was able to use this control to impose a new development paradigm onto all developers. According to Nicholson and Sahay, British management viewed the software development process as machine-like, and saw developers as interchangeable parts, a view with implications for long-term issues of job satisfaction, employee aspirations and company loyalty.

Nicholson and Sahay illustrate that distributed software development needs to be understood within its full social, cultural and political framework. Their study foregrounds the impact of national and cross-cultural issues on distributed development projects, an important point as global outsourcing arrangements usually draw diverse cultural contexts together.

Kobitzch et al. (2001) describe the experiences of an outsourcing arrangement between a German telecommunications company and a software development organization located in Bangalore, India. From this, they develop a systematic overview to aid companies considering outsourcing projects. They suggest that, despite the multiple cost overheads that characterize distributed software projects, distribution can solve problems of qualified personnel shortages, and can facilitate access to lower development costs and new markets. Corporations interested in outsourcing have four models to choose from: (1) separate teams operating in independent companies; (2) separate teams operating in legally related companies; (3) one team dispersed across multiple sites of legally related companies; and (4) one team dispersed across multiple sites of independent companies. The authors contend that which of these models an organization adopts will depend on the answers to a number of questions. What historical relations exist between the companies? Are there any cultural boundaries that need to be considered? Along what criteria will development be distributed (i.e. systems requirements, development cycle, expertise)? Kobitzch

et al. argue that the answers to these questions are important, as no single approach will satisfy all situations.

They suggest further things to consider, such as language differences and "temporal dispersion (Kobitzch et al. 2001:82)." Models (1) and (2) are relatively robust in the face of language differences, while models (3) and (4), entailing extensive intra-team communication, are very sensitive to language difference. Similarly, even small time-zone separations can have significant impact on a project. Not only are there often difficulties caused by the reduced work-overlap caused by time-zone difference, but further temporal separation is introduced by local holidays and work patterns. Once again, the models are varyingly sensitive to these factors.

The authors report two critical factors for success that emerged from their experiment in global software outsourcing. First, tight management control (project schedules, deliverables, reporting and risk assessment) must be given priority. Also, personal relationships must be nurtured through whatever means are available (e.g. videoconferencing, advanced communication infrastructure, and frequent site visits).

Heeks et al. (2001) suggest the strategy of *synching* as a way of preventing global outsourcing arrangements from *sinking*. They use synching to describe the successful relationships between organizations, relationships characterized by "a high degree of congruence...between developer and client (Heeks et al. 2001:55)." Sinking, as a description of an organization-to-organization relationship, is somewhat self-explanatory. The researchers use the term congruence to refer to alignment along six dimensions: "**c**oordination/control systems, **o**bjectives and values, **c**apabilities, **p**rocesses, **i**nformation, and **t**echnology (Heeks et al. 2001:56)." They used these *Cocpit* dimensions to study outsourcing arrangements involving Indian companies and found that synching is a considerable challenge:

> Western clients seem to fall too easily for the argument that, in a globalized world, distance, borders, and place no longer matter. The experience of these cases suggests otherwise. (Heeks et al. 2001:58)

One of the biggest challenges can be framed in terms of *tacit knowledge*. Michael Polanyi (1966) was among the first to use this term to describe what we know, but may not be aware *that* we know or *how* we know. Tacit knowledge can be contrasted with explicit knowledge, which "can be laid out formally (Heeks et al. 2001:58)." Heeks et al. argue that, in global outsourcing projects, managers need to actively attend to tacit knowledge in all of the 6 *Cocpit* dimensions as it has a substantial impact on the success of synching strategies. For example, cultural differences – based in subtle practices and beliefs that people are often not aware of - can present barriers to the synching of objectives and values. These barriers can be lowered by bringing people from the different organizations together, and allowing them to get to know one another, thus synching tacit knowledge.

Kavanagh and Kelly (2002) also argue that place matters, and cannot be glossed over by the rhetoric of globalization. They studied an international pharmaceutical corporation designing and developing a new process control system for a plant in Ireland. The authors look at the networks involved in this project, and in particular are concerned with how the various actors act politically within these networks. Ultimately, they maintain that knowledge of space and location are fundamental to understanding how complex multi-national, multi-organisational, knowledge-intensive networks function.

Newell et al. take up this argument, and position themselves to "dispel the myth of the boundaryless organization" and argue "against technologically deterministic approaches to organizational change" (Newell et al. 2001:74). They observe that many global organizations view collaborative technologies as holding the promise to "overcome the twin tyrannies of spatial and temporal dislocation that impede global reach" (Newell et al. 2001:74).

The subject of one of their case studies was a global bank that had become global through the acquisition of several smaller banks. Managers decided to use the internet to create a true global organization, and believed that by implementing such a collaborative technology, the various boundaries within the organization would dissolve. Instead, the researchers found, the boundaries were recreated as "electronic fences" (Newell et al. 2001:75) when each of the various

divisions designed and implemented their own completely independent intranets. Another case study involved a global chemical business that wanted to dissolve global divisions by creating an internet forum oriented toward knowledge sharing. According to Newell et al., divisions within the online forum soon emerged on their own, as people began to establish and post to geographically bounded groups. The response of management was to install skilled facilitators to act as moderators for the online groups, and to reorganize the forums according to industry. The result was a dynamic and powerful knowledge-sharing network.

The researchers argue that these two case studies show that potentially collaborative technologies cannot be invoked to cause organizational change and erase internal boundaries. They also maintain that some boundaries within organizations can be effective, acting as aids to globalization, and should be encouraged, not eliminated.

Maybury describes an instance of collaborative technology in which, defying these cautionary tales, distributed teams did manage to "overcome both time and space challenges" (Maybury 2001:53). The subject is a collaboration tool (a *collaborative virtual workspace*) designed to facilitate air campaign planning activities in a national defence context. Users of the system, distributed across a wide geographic area, use it to plan and enact complex air operations exercises. According to Maybury, the system dramatically improved their task performance. What this example illustrates is how well collaborative tools can work when the task is very well defined, the users of the collaborative tools are united toward a single goal, and the tools are deployed in an environment where people are most likely accustomed to working with others in a distributed fashion.

Many researchers have begun to consider in detail the *goings on* (Orr 1996) of distributed work. As far as my research is concerned, Herbsleb and Grinter (1999) completed one of the most important projects. They studied a software development project that was distributed between the U.K. and Germany, and found that the key issues were related to coordination and control. According to the researchers, there is an inherent unpredictability to software

development that cannot be adequately addressed through process. The solution is to ensure that capable workers are employed, and that there is certain flexibility in the way work can be done. Herbsleb and Grinter submit that barriers to communication dramatically affect the success with which unpredictability is managed. They observed problems with conflict resolution, a lack of trust, failure to work toward a common goal, and frequent feelings that the parties in the other office were being stubborn. They noted situations where changes to code that should have been trivial were stretched out over several days. Herbsleb and Grinter's results so strongly parallel my own, that the reader will be surprised when I insist that it wasn't until after I did all of my fieldwork and most of my analysis that I discovered their paper. Many of their findings will be invoked in greater detail in a later chapter.

Sproull and Kiesler 1991, O'Mahony and Barley 1999, and Olson and Olson 2000 have investigated how specific technologies affect distributed work. Sproull and Kiesler studied the dynamics of groups and found that groups constituted electronically behave very differently from face-to-face groups. They note that electronic interactions are far less sensitive to status than face-to-face interactions, that electronic groups are far more likely to take extreme decisions, and that flaming – directing abusive remarks to another individual – is substantially more common in electronic groups. In addition, they learned that electronic groups make decisions quite differently from face-to-face groups. More people are consulted in electronic groups, and the contributions from people who exhibit faulty reasoning, but who have excellent face-to-face communication skills, are more likely to be disregarded.

In an extensive review of the literature concerned with the technologies of distributed work, O'Mahony and Barley (1999) found that group decision support and electronic meeting systems are good at collecting input from a wide variety of individuals, but less effective when tasks are ambiguous or poorly defined. They note that where video has a positive impact on group cohesion, it usually supports existing relationships better than it creates new ones. A similar thing can be said for e-mail: it tends to broaden group participation only where existing group ties already exist. In general, O'Mahony and Barley state,

technologies are seldom the transformative force that they have been promoted as being. Instead, social and cultural factors usually shape the way in which technologies are adopted.

Drawing on a decade's worth of their own research, Olson and Olson (2000) state, "to paraphrase Mark Twain, the reports of distance's death are greatly exaggerated" (Olson and Olson 2000:140). They suggest that, not only has distance not been mastered yet, it will never be completely mastered, and they support this claim with a comprehensive summary of their own empirical research. In order to simplify the problem, they restrict their attention to work that considers distributed workers who communicate mostly through synchronous means.

With respect to video, Olson and Olson suggest that we don't know enough yet to proclaim conclusively on the success of this communication medium. They note that although data previously supported the claim that video didn't add much to distributed work effectiveness, things seem to change when multiple speakers who do not share a native language are introduced. Although people communicating in a native tongue do not seem to get much extra from video, others find that video facilitates comprehension and understanding. Olson and Olson report that poor audio or video quality, or slow application sharing facilities can lead to frustration and complaints.

They suggest that the social contexts in which tools are deployed are important to successful adoption. Tools must be aligned with organizational expectations and rewards. For example, if companies want to adopt collaboration technologies, then employees must be rewarded for collaborating, and for using the technologies. Where there are incentives to collaborate, people will adapt to even the most awkward and clumsy tools. Olson and Olson argue against technologically deterministic claims that people will use the tools that communicate the most and best information. Instead, they maintain, the reasons for adopting particular tools are just as often symbolic as they are functional.

Other researchers also argue that distance and time matter in distributed work. Grinter et al. (1999) suggest that it is the coordination part of collaborative work that suffers most from distribution. They argue that this is because much of

this coordinating work relies on informal communication channels, and that these channels function poorly when teams are dispersed. Herbsleb et al. (2000) point out how delays can plague work that is distributed across multiple sites. Part of this delay can be attributed to the reduced workday overlap that is inherent to projects that involve people in multiple time zones. Another contributing factor is the difficulty that people have in figuring out who they need to talk to at the remote site. One of their more interesting findings relates to how much workers perceive colleagues at remote sites as helping out when the workload is heavy. According to the researchers, people frequently believe that they help remote colleagues as much as they help local colleagues, but simultaneously believe that remote colleagues are much less helpful than local colleagues.

Kraut et al. (2002), and Kiesler and Cummings (2002) discuss proximity, and its effect on the collaborative process. Their research results show that engineers and scientists who are collocated are much more likely to undertake collaborative projects than they are with colleagues located in a different geographic location. This is the case even in a network with easy access and to and extensive familiarity with electronic mail tools. The researchers suggest several possible explanations. Certainly it is easier to communicate with someone in the same office, and in particular there is a much higher opportunity for chance encounters. They suggest that these chance encounters establish significant common ground between individuals: they have shared experiences and knowledge, leading to enhanced communication.

> Each communication episode provides the potential for people to learn something new about their partners, make decisions, monitor the state of the work, take corrective action, and perform other joint activities. If the communication episode does not take place, then the information exchange and joint action will not occur. (Kraut et al. 2002:141)

Kraut et al. consider various media, and the degree to which those media capture the character of communication that appears necessary for collaboration. They find that the chat room seems to be one of the most effective choices, better even

than a continuous audio/video link. It has a good blend of spontaneity with accessibility, facilitates the fostering of new relationships, and provides privacy when it is needed.

Kiesler and Cumings observe that when people work in each other's presence, their attention is increased, they have greater social impact on one another, and they are more familiar with one another. Conversely, the lack of co-presence leads to reduced attention to co-workers needs, lessened social impact, and decreased familiarity – "out of sight, out of mind (Kiesler and Cummings 2002:63)." People are usually busy, and responsible for the completion of many tasks. If some tasks involve an absent co-worker, those tasks may be given lower priority.

On the other hand, Kiesler and Cummings note that that collocated work can be highly disruptive as people are vulnerable to interruptions. They suggest that in cases where the work to be done is relatively independent, and working alone is appropriate, distributed work may in fact produce better results.

Kiesler and Cummings argue that casual contact is important to relationships. Distance severely limits spontaneous, informal contact. Those who are collocated run into each other at the coffee machine, leading to familiarity and more pleasurable communication.

> With spontaneous casual communication, people can learn informally how one another's work is going, anticipate each other's strengths and failings, monitor group progress, coordinate their actions, do favors for one another, and come to the rescue at the last minute when things go wrong. (Kiesler and Cummings 2002:66)

They suggest that where strong group cohesion already exists, communication technologies may work well at facilitating distribute work.

From the above presentation, it should be clear the considerable research attention has been given over to the problem of distributed work. Perhaps one of the more common refrains throughout this body of literature is that distributed work is complicated. No one tool can fix everything; no two situations are alike.

Much of the above work will become directly relevant to the analysis of my own research results.

3. Method

3.1. Setting up the Project

From the outset, this research project has been an exploratory one, directed toward determining those factors that characterized K2 distribution by drawing on the varied experiences of the project's participants (in the interests of maintaining participant anonymity, pseudonyms have been used for the company and the project: these pseudonyms are Chimera and K2 respectively). To that end, the best research strategy was a qualitative case study.

The exploratory nature of my research made it ideally suited for a qualitative study. Qualitative approaches are appropriate for research conducted in a field not yet extensively explored by other researchers. They are also useful when the research is concerned with issues of process, meaning or understanding, all of which are important for my project. Finally, qualitative research does not typically engage in hypothesis testing. Instead, theory and interpretation are expected to emerge from data collection and analysis as the qualitative researcher takes an inherently inductive approach to her project (Creswell 1994).

Several data gathering techniques comprise the toolkit of the qualitative researcher (Creswell 1994): observation notes, participant-observation notes, unstructured interview notes, audio recordings of unstructured interviews, field journals, and the collection of correspondence or public documents produced or used by research participants. Those specific data collection techniques adopted in any particular research project will depend on the nature of the research question and, as importantly, on the particular 'real-world' conditions at the research site. In the case of this project, these 'real-world' conditions played a significant role in the data gathering techniques that I ultimately chose. A brief overview of the history of my relationship with Chimera will clarify these choices.

Initial discussions among company representatives, my graduate advisors, and myself revealed several promising prospects for distributed development research. The company has at least five international software development sites

and has undertaken many geographically distributed projects. One project in particular emerged as the most likely research candidate.

A senior manager in the company who had taken an active interest in my research was about to embark on a new role as manager of a project that was to be distributed between at least two of their international offices. This promised to be an excellent research opportunity as I would be able to observe and perhaps even take an active role in the new project. A project team was to be assembled and training provided under the management of an individual keen to welcome the participation of an outside observer.

Preliminary discussions in the spring of 2001 set the stage for this project, and I proceeded to review the relevant literature and develop a proposal that described my intended research approach. According to that proposal, I would collect data through four main channels. First, there was the possibility that I could fill a useful role in the project and function as a participant-observer who would spend significant time in both offices. In that role I would have access to two more sources of data: artifactual material produced as the project progressed, and my own detailed field notes in which I would capture my reflections on the fieldwork experience. The fourth and final source of data would be a series of unstructured interviews with project participants, to be conducted at various stages of the project.

During the six months following those preliminary discussions, events conspired against this approach. The first major shift was a declining economic position at Chimera. The summer of 2001 was a turbulent one, particularly where high-technology firms were concerned. This general economic uncertainty was exacerbated by events of September 11 in the United States. During that six-month period, the company experienced substantial contraction with the dismissal of a significant number of employees. Those in upper-management were not inclined to support a research project that they perceived would provide little direct value to the company.

There were other developments during this six-month period: two major personnel changes occurred and each had significant impact on my project. First, the manger slated to lead the newly forming distributed project was transferred

to an entirely different position. When this happened, it was not clear how the software project that I had planned to study was going to proceed, or if there would be any role for participation of an outside researcher. There was a further complication when the person at the company who had first made connections at the university regarding possible research collaborations, resigned. I found myself in a position where I was pursuing research in very tenuous circumstances.

Ultimately, the original research project became unavailable to me, and I was faced with two options. I could find a new company with which to start over, or I could rework my project to fit new opportunities within the company that I had already worked with. In consultation with my advisors, I opted to pursue the second path. My attention shifted to an ambitious venture in which a software development team was split between a city in Canada and a city in Germany. This spatially distributed team had worked for two years on a tightly coupled and highly complex product: an excellent candidate for study.

The primary complication with this second research project was that the distributed aspect of the software development had been terminated in May of 2001. The team was no longer distributed as all work had been consolidated in the German office. Thus I would need to base my study on a project that was already over. Originally, I was going to be involved from the earliest stages in a new instance of distributed software development. In the new, reformulated project, I was to gather information covering a two-year period during which a major product development team had been distributed across two international offices.

The dramatic transformation in the nature of the available case study necessitated a major change in data collection methods. In order to best access the available information, I used three data collection methods. The primary source of information was semi-structured, open-ended interviews with individuals who had participated in the distributed development process at the company. In order to get the broadest possible range of perspectives, I asked two senior managers to compile a list of names representing people from a variety of positions in the company. Throughout the process of developing the study, these

managers were always helpful. They gave me no reason to believe that names of potential participants had been strategically chosen in order to reflect a certain position. On the contrary, I view these managers as research collaborators truly interested in considering the subject in detail. They provided me with more names that I could use, and the list included people who had left the company.

Our primary goal was to capture those individuals who had been involved in K2 during the relevant period of time. Many of those individuals were on the resulting list, as were others who had been only peripherally involved in K2 but who had had extensive involvement in other distributed work at Chimera. The interview data contributes to a picture of how distributed development happens at the company in general, with detailed reflections focused on one project in particular.

Some researchers point to certain challenges with interview data, particularly when interviews are the only source of data in the project. Interviews, after all, collect what people experienced (Marshall 1995). In an exploratory study such as this, the reflections, interpretations and opinions of the research participants are a valuable means of constructing a picture of how events transpired. It is critical to note that I am not concerned with evaluating the truth value of the contributions of interviewees - determining who reported the *real* story. This case study is an attempt to identify certain characteristics of what was a complex process and there will always be different experiences to emerge from such a process. In fact, this multi-perspective aspect of the project is part of what I am interested in. Representing a variety of perspectives and using supplementary data to triangulate and fill in the blanks can mitigate any limiting aspects of interviews.

One such source of supplementary data is my field notes. Throughout the development of this project and during all time spent in *the field*, I have kept detailed notes that include not only direct reflections on the events related to interviews, the selection of interviewees, notes from informal meetings, and my experiences in traveling to various field sites, but they also contain my own personal reflections on the fieldwork process. These notes situate the interviews in a particular context, retaining a sense of not only the ephemera relating to the

interviews, but providing a sense of my own mental state. This contextual information is helpful in interpreting the interview data.

Finally, I retained all correspondence relating to the set-up of my research project. In conjunction with my field notes, this record, which is entirely in the form of e-mails, is useful in providing a broader context for the research project itself. It provides a picture of the community in which the research project is situated, particularly the volatility and uncertainty that the researcher engaging in industrial or organizational research must acknowledge, anticipate, and prepare for or, at the very least, adapt to.

3.2. The Interviews

In total, I interviewed seventeen people at Chimera. The length of the interviews ranged from fifty minutes to ninety minutes, but averaged approximately one hour. Selection of interview participants happened in two phases. The first phase occurred early in the process, while it still appeared that I would be doing an in-depth study of a distributed project as it developed. At this time the senior manager who had been my main contact at the company suggested several names of individuals who would be able to provide me with some initial information regarding their experience in doing distributed development. These individuals did not share a common participation in any one distributed project, but had experience spanning many business areas within the company. I interviewed these three people in November of 2001 and saw those interviews as a preliminary step in preparation for the later, larger research project.

The next interview phase developed when the project scope had been changed and was now concerned with examining the development of K2 primarily between October of 1999 and May of 2001. The manager responsible for my continued involvement in research at the company as well as one of the managers who had been involved in the distributed development project at the center of the study created a list of individuals likely to be an interesting source of information. This list included technical personnel such as programmers,

database administrators and team leads, as well as business analysts, trainers and managers.

I was given the list, and asked to delay making contact with people until such a time that both management and potential study participants had been made aware of the project and that I might be contacting them. This process took several weeks as each time I inquired if I could go ahead and start contacting people I was told that they had not had an opportunity to talk to everyone yet. In addition, we had to engage in some last minute crisis management: even at this late stage of the game, it was not clear that I had approval from the necessary senior managers to proceed with the study. Throughout my fieldwork, my position as a researcher in the company was in no way secure and I was continually aware of the need to renegotiate my presence.

Finally, I was given permission to contact people and arrange interviews. This happened in April of 2002. I wrote an introductory message introducing my project and myself (a sample is included as an appendix), and sent this via e-mail to each of the potential research participants. I decided that e-mail was a suitable means of establishing communication because the company is an international software development company that has adopted e-mail as a primary form of contact. While e-mail is a highly problematic form of communication, as I found in this research project, it is suitable for exchanging information that is relatively straightforward. I adopted a formal tone in the e-mail message, much as I would have done if I were sending a business letter. By initiating contact in this way, I left it up to the individual whether they wanted to participate in the study and respond to my e-mail or ignore my message and thus choose not to participate.

I sent nineteen e-mail messages. Of those nineteen, two people never responded and two responded late, once I was no longer in a position to interview them. In addition, one person was removed from the list when I was in Germany because nobody could figure out why she'd been put there in the first place. Thus out of the nineteen exploratory e-mails and one face-to-face interview request, fourteen interviews were successfully conducted in the second interview phase.

All interviews happened either in Canada or in Germany. The Canadian interviews were conducted during two periods: November of 2001, and May

through July of 2002. I asked each research participant to choose where he or she wanted to conduct the interview. There were several options available to us. I had been granted access to any of the company boardrooms and volunteered to book them as required. For those individuals with a private office, I suggested their office as an acceptable place for us to meet. Finally, the company office is located in a part of the city with a good selection of coffee shops, so I suggested they might want to meet over coffee or tea. Of the ten interviews conducted in Canada, four people elected to meet over coffee, three decided to use their own office, two wanted to meet in a boardroom, and one chose to meet in an office away from company property.

All of the German interviews were conducted during a nine-day period in the middle of May of 2002. As was the case in Canada, several interview locations were available to us and again I asked the participants to choose their preferred location. Each floor of the German office contains a small meeting room that anyone can use. This room is available in part because everyone involved in software development at the German site shares an office. Another option was the company cafeteria located on the ground floor of the office building. The third available meeting place was the common room of the company apartments, where I stayed during my time in Germany. These apartments are located in a building that is connected to but still separate from the software development offices. On each floor are six small apartments with a shared kitchen and a shared common room. For the duration of my stay, this common room was usually empty and it turned out to be an excellent meeting place. Of the seven interviews conducted in Germany, three were conducted in the common room of the company apartments, two took place in the individual's office, and one was conducted in each of a meeting room and the company cafeteria.

I began all interviews by talking about the interview consent form (attached as an appendix). By going through each point on this form, the interview participant was explicitly informed as to my status as the researcher, the objective of my research and how I planned to use my data. They were given the opportunity of choosing not to participate or to terminate their involvement

in my study at any time. Finally, they were assured that all of their responses would remain confidential, and only reported in my research publications in an anonymous form. This form was written in English, and I offered all German participants the option of taking it away to study it more carefully.[1] None elected to do this, and all participants signed the consent form.

Next, I asked if the participants would allow me to tape-record our conversation. I stressed that by recording the interview, I could gather more information and attend more carefully to the discussion without having to worry about taking detailed notes. I also stressed that the recordings would be mainly for my use, and that if I used a direct quotation from the transcripts in any published material, I would consult them first. Nonetheless, three people asked that I not tape the interview, and in a fourth case, it was not practical to tape. For these four interviews, I took very detailed notes, and recorded my own impressions on tape immediately following the interview.

For the first few interviews, I created a specific set of questions that I worked through in the course of the conversation. As I conducted more interviews, this set became smaller, and the individual questions became more general. I found that the best approach was to use the responses of the individual to guide the course of the interview, relying on my prepared questions to elicit certain specific information or to invoke when the conversation lagged. A sample of the questions is attached as an appendix.

As a final step in data collection, I transcribed the audiotapes of the interviews in order to render all interview data in a consistent textual format for systematic analysis. The seventeen hours were transformed into over three hundred pages of interview transcript. I elected not to transcribe my field notes as they are interleaved with sketches and diagrams made during interviews. Their hand-written format does not preclude their availability to analysis.

---

[1] In retrospect, I should have had the interview consent form and a list of prospective questions or areas of interest translated into German.

3.3. Data Analysis

One of the more challenging aspects of qualitative research is the volume of data that is created and the difficulty of organizing those data into a manageable form (Marshall 1995, Creswell 1994). In an exploratory study such as this, where data are not extracted in order to test a particular hypothesis, the role of data analysis is to reveal patterns in the interview data and make categories (Creswell 1994). To that end, I have adopted and adapted a general approach described by Creswell.

The first step of this approach is to review all transcripts, noting down observations and ideas as they arise. I found the transcription process, while tedious and painstaking, an excellent opportunity to reflect upon the interview, recalling what was said, and reflecting upon what was said in the context of other things. As I typed the text into the word processor, I used the comment feature to note down things that occurred to me at the time. Later, I reviewed these comments and transferred them to printed copies of the transcripts. I also reviewed all field notes and supplementary data. There were two main outcomes of this first phase. First, I refreshed my memory regarding events that had taken place over a span of several months. Second, I began to consider the complete body of data rather than the distinct parts. From this first phase I created a preliminary set of topics, those categories that emerged from a preliminary review of data.

I devised a simple set of codes to reflect these preliminary categories and re-read the transcripts, marking down the appropriate codes next to the relevant sections of text. Not all text has a code associated with it, and some sections of text are marked by multiple codes. After reviewing the interview transcripts a second time, I further refined the codes: in some cases I expanded them into multiple new codes, and in others, I eliminated or collapsed them into other, existing codes. Finally, I did a rough summary to see the representation of each code in each interview.

It is important to note here that I did not create codes in order to apply

any sort of quantitative analysis. Nor did I employ a sophisticated qualitative analysis software application in order to undertake a rigorous analysis of the text. Instead, the codes serve a heuristic function for me, the researcher. They were a way for me to somewhat systematically engage with the data in order to identify and understand emergent patterns in what people had to say.

I created and applied twenty-four codes in total. Three of these codes marked sections of the interview transcripts that provided general company facts, details on a broad array of products that they produced, and information on the specific product that many of the research participants had worked on. This forms much of the background information and context presented elsewhere in this work, but is not separately presented as data; nor is it part of the analysis phase of the project. This leaves twenty-one codes that I used to systematically process the interview data. The results of this systematic review process are presented in the next chapter.

4. Observations

4.1. An Overview

This chapter contains a detailed summary of my observations, amalgamating field notes, interview notes, and interview transcripts. They are presented here with limited interpretation on my part: for now, at least, the words of the study's participants will have to speak for themselves.

There are, of course, certain vital ways in which nothing can ever really "speak for itself". First and foremost, I have included only a subset of all that was said in the many conversations from which the transcripts were created, presenting only certain key excerpts that I believe illustrate the main ideas that were discussed.[1]

Additionally, in order to provide a comprehensive set of observations without resorting to a tedious succession of similar interview excerpts, there are many cases where I summarized participants' responses, or where I allow the words of one individual to stand for several who made a similar point. While I have made every effort to preserve what I saw as the essence of what was expressed, I acknowledge the impossibility of actually achieving that goal.

The summarized observations are presented here in a format determined by my own codes. This is just one possible ordering, which itself could have led to diverse forms. The results are complicated and do not lend themselves to tidy categorization. Many excerpts had multiple codes affixed to them, placing them into multiple categories. In some cases, the category selected for presentation purposes was chosen because the text represented the respective theme or practice particularly well. In other cases, the final choice of category was somewhat arbitrary as each theme or practice identified with the segment was illustrated with equal weight. Consequently, there are many cases of overlap,

---

[1] With over 300 pages of transcripts, I doubt that even the most avid reader would want to plough through the entire body of interview data. Indeed, it was an imposing undertaking.

examples that illustrate the same themes or practices scattered throughout many sections of this chapter.

In total, I applied twenty-four codes to the data. Three of these reflect general information about Chimera and the K2 project, and form much of the background information presented elsewhere in this work. The remaining twenty-one codes have been organized into three categories. The first category contains the five *themes* of distributed development identified from the participants' responses: time, language, culture, power and trust. I used these codes to identify all of the passages in the transcripts concerned with these abstract or broad conceptual challenges to distribution. The second category consists of the various specific ways of doing things on the ground, on a day-to-day basis within the company, as described by the participants. These include the use of particular tools or systems, as well as choices about how to conduct the project. Adapting the work of Orr (1996), and Brown and Duguid (2000), I chose to call these specific ways of doing things *practices*, and argue later that it is at the level of practices that the themes are realized and reinforced.[2] The twelve practices identified in the data are itemized in Table 1.

Table 1 - Practices

Practices		
Communicating via e-mail	Other tools	Group size
Communicating via Telephone	Software Process	Individual personalities
Teleconference meetings	Experience level of developers	Management style
StarTeam system	Proximity of developers to one another	Project scope

[2] As my project was based on interviews rather than on extensive participant observation, I cannot comment on the specific day-to-day activities, as recorded by the researcher, that Orr, Brown and Duguid define as practices. Nonetheless, this term captures the distinction between the broader contextual factors – the themes – and the actual workplace activities that participants described.

A third category contains general observations made by the participants about the process of distributed development. This category contains the participants' broad reflections on the overall problem of distribution, not necessarily specific experiences. There are four codes that make up this category: comments about communication at a very general level, discussions of the economic cost of distribution, ideas about why the company would even bother to pursue distributed development, and observations about some of the characteristics of collocated development projects within the company.

The remainder of this chapter presents the participants' responses, starting with the five themes, followed by the twelve practices, and concluding with the general comments regarding distributed software development.

A few notes on representation technique are required. All participants in the study were guaranteed anonymity. In order to honour this guarantee, the names of individuals are not provided along with the interview excerpts, or anywhere in this work for that matter. Instead, an interview code is provided in square brackets along with the excerpt, and only my graduate supervisors and I have access to the mapping between the codes and the identities of the interview participants.

For the same reason, the personal pronouns that accompany the presentation of observations do not necessarily reflect the gender of the individuals who provided the responses. Sometimes a *he* really is a he, other times I have used *he* when a woman provided the comment. Similarly, a comment was not necessarily provided by a woman simply because I wrote *she said*. I chose to take this approach rather than bend over backward keeping all pronouns, which might identify the interview subject, out of the text.

Finally, it was my experience that all interview participants were reluctant to name names, particularly when they were relating negative experiences. In several instances, people about to mention a specific person caught themselves, and continued relating their experience without identifying the person. Thus the seventeen interviews, excerpts of which are dispersed throughout this chapter, should be considered to some degree as individual stories as there is little specific information to connect particular events across multiple interviews. Even though

there are no specific common events that we can isolate from the interviews, participants reported many similar experiences with distributed development.

4.2. Themes

As stated earlier, themes represent the more abstract or conceptual challenges or characteristics of distribution. Invoking the information ecology metaphor, these make up the inputs to the local system, or the constraints placed on it. An equivalent in an ecological system might be available sunlight or precipitation. I noted five themes in the data: time, language, culture, power, and trust.

4.2.1. Time

> *It's not only that it's distributed; it's distributed by time, so that you...can't see each other, and you can't talk to each other really, because one party is always tired as the other party starts to work. [18]*

There is an eight-hour time difference between the two offices, and most people mentioned the intractable problem presented by this temporal separation. Consequently, time emerged as a theme early in the data collection phase. This category includes comments about the impact of temporal separation on any aspect of the process of distributed software development.

One of the most straightforward aspects of the K2 project was database administration. There were only two people working as database administrators (DBAs): one in Germany and one in Canada. Their roles were clear, and what and when they needed to communicate were well constrained. Once a good communication relationship between these two individuals was established, the distribution of the database administration was relatively unproblematic [11]. The only significant challenge to the distribution was the eight-hour time zone difference. Significant delays in troubleshooting were encountered whenever problems requiring the other DBA's input were encountered; the time-zone factor rendered these delays inevitable.

Similarly, a participant discussed the time-zone problem in the context of a project that had originally been distributed between two Canadian cities. The project had been successfully operating in a distributed fashion for some time when it was further dispersed to Germany. The extended distribution worked out reasonably well with the exception of one persistent challenge: "The biggest difficulty once again came with the time differences. If something went wrong with the compile, then there was usually about a one day delay so that would bugger up the schedule" [4].

In these two examples, while the challenge introduced by the time zone separation was indeed frustrating, it was not an insurmountable barrier to continued distribution. Other aspects of the project worked well, and ways of dealing with the eight-hour time difference were devised. This contrasts sharply with the experience reported by participants in the K2 project.

Overwhelmingly, people who had worked extensively on K2 reported the temporal separation to be a severe obstacle to successful distribution; an obstacle that many indicated was reason enough not to distribute at all. Perhaps the most oft cited explanation for why time zones were so problematic was the required reliance on asynchronous communication techniques[3] and the overall negative impact that this had on problem solving. The following description given by a developer illustrates the impact that the time-zone difference had on the sort of issue that would be easily resolved with a five-minute conversation:

> Taking the time difference, if I come to this realization at noon on a Wednesday, you don't see it 'til the next day, because now it's midnight your time. You'll mull it over for the day, get a little pissy about it, send me back your *angrygram* sometime around four o'clock your time, which is just when I'm rolling into the office. The odds of my catching you on the phone are pretty thin, unless you're working really long hours... I have maybe that one hour or half-hour window where I've got to catch up, and maybe you're forty-five in a list of a hundred e-mails. I've got to get up to yours and read it, realize the problem and call you. Otherwise, all I'm doing is sending another e-mail, restating my point with some other mitigating factor, and waiting for you to do the same thing over again.

---

[3] Synchronous communication describes participants who engage with one another at the same time. Asynchronous communication does not coincide in time.

> You've just stretched a five-minute deal into a week...And there's nothing like letting it simmer for an evening to really get somebody choked about it and antagonistic for the next day. [14]

A senior manager indicated that there were few technological solutions available to address this problem, and that the best approach was to limit what had to be communicated:

> One of the biggest areas we can talk about...was just the time zone difference, just a huge variance...Even if you have giant pipes[4] and could communicate whatever you wanted you've still got a fairly narrow window that the team can communicate. Because of that, even in an ideal circumstance, you need to choose what you're sharing very carefully. [4]

Others indicated that the time-zone problem was the one challenge to distribution that was insurmountable: "I would say that language is not an issue. Language is something that you can fix with training. The time zone is something that you can't fix" [18].

While many people talked about the separation in time between the two offices, nobody talked about the enormity of the physical separation. Nobody mentioned that the other office was thousands of kilometres away, or discussed how difficult it was to get to the other office except in the context that travel was expensive, and thus likely to be the victim of budgetary reductions. They did talk about the importance of proximity to successful development, but suggested that in this respect, even a small geographic separation is a problem.

4.2.2. Language

> *When people who haven't been over to Germany speak to people in Germany, they'll use a lot of English idioms and things like that. It just makes it harder to communicate, and people in [the German office], most of them speak very good English, but, it's just - some of the minute differences in their choice of words makes their decisions a little more ambiguous. And because of the friction that's happening between these*

---

[4] 'Giant pipes' refers to the capacity of the computer network connecting the two offices. A high capacity, fast network – a giant pipe – facilitates very efficient transfer of data.

two offices, whenever there's any ambiguity in what they're talking about, they're usually not given the benefit of the doubt because of that friction. [9]

Like *time*, *language* emerged as a theme early on in the data collection process. This category includes participants' comments concerning working in a multi-lingual environment in which English is the common language and many of the speakers have something other than English as their native language.

My first inkling that language was an issue in the organization came when I began to arrange the German interviews via e-mail. I was assured by most of the people who I had spoken to in Canada that everyone in the German office spoke English very well, and that I would experience no difficulties conducting the interviews in English. I sent a message via e-mail to all prospective participants, introducing the project and myself. That message was entirely in English. Included in almost every response from potential participants situated in the German office was a request for a list of questions. I found this remarkable because none of the Canadian participants had asked for a list of questions, leading me to believe that perhaps some of the people were uncomfortable with the interview process, particularly when it was to be conducted in English.[5] That this was the situation, at least in some cases, was confirmed when several people showed up to do the interviews armed with English-German dictionaries.

Many of the German participants indicated that there were situations in which they encountered difficulties because of the requirement of doing everything in English [10,12,16]. They did not suggest that things should be done any other way as English is not only the language of the corporation, but also the language used for communicating with clients. As noted by an individual from the German office, it didn't make sense to work in anything but English: "We don't have a German client, so it's English. There's no other choice" [12].

---

[5] I do not mean to suggest that this is the only conclusion to be drawn regarding the request for questions, only that this was a possible explanation.

Nonetheless, many participants reported that working in their second, or in many cases third, language was challenging.[6] For example, German employees expressed a preference for receiving technical training that was conducted in German. Initially, there was a relationship between Chimera and a Canadian university. Professors from the university travelled to both the Canadian and German offices and provided uniform technical training. While this training was generally well received, a new relationship was created between the German office and a local German technical university and training is now provided in the language of the local office: English in Canada, German in Germany [7].

The most challenging situations reported by German speakers who worked in English were those requiring greater than average concentration. For example, a manager who appeared to me to be completely fluent in English stated that long business meetings were often difficult because fatigue makes it harder to function in a second or third language [16]. Another person indicated that situations involving conflict or argument were demanding [12]. Most of the participants from the German office noted that they found participation in teleconferences to be particularly challenging as they often could not understand what was said in the other office. Many Canadians who had spent time in the German office echoed these observations, stating that they had witnessed this difficulty first hand [4,5,9,13].

Not only was comprehension on the part of the Germans a problem in teleconferences, misinterpretation on the part of individuals in both offices often introduced or exacerbated conflict in the meeting. The following comment from a member of Canadian office who had spent time sitting in on meetings in the German office illustrates this phenomenon:

> I'd be sitting in there...the Canadians would phone in and they would have this discussion and somebody would say something the Germans would take horrible offence to because it was improperly worded or something. And I'd say, "No, no, no. They didn't really mean it this way. They meant

---

[6] Many of the employees in the German office grew up in what was formerly East Germany. Consequently, they would have learned German and Russian, making English a third language.

> this."...And they'd go, "Oh. Alright." And then they would say something back and the Canadians, "You rude bastard!" [14]

Language problems were not restricted to incorrect translations of words, a participant who had spent considerable time in both offices noted. Instead, he suggested that linguistic difference included different ways of thinking that went hand in hand with the different words [6]. Accordingly, miscommunication between the offices was complicated and took many forms. Often, terms would be employed that would have very different interpretations in the two languages. German speakers would frequently choose an English word with a different meaning than they intended. Several individuals indicated that when they knew the speaker personally, it was easier to know what they meant to say and when they'd chosen the wrong word [6,13].

A multilingual participant from the Canadian office indicated that at times, when he was working in the German office, he felt like Switzerland[7]:

> I understand the people there, and I know them personally. Also, I kind of know what they're trying to say...Somebody would complain and say, "Look at this e-mail! What do you think about this?", and I'd go and I'd translate it into German and go, "Oh, you know the only thing they're saying here is...", and so on and so forth. [13]

It was a common perception, related by participants from both offices, that Canadians generally underestimated the scope of the language challenges that German employees faced [5,6,13]. According to one participant, there were individuals with particularly bad attitudes within the organization: "We've got people that, if you say something to a German once, and they don't understand it, then you yell at them, figuring that they're stupid" [5]. Alternatively, I spoke to people who were sympathetic to the challenge that language presented to those in the German office who did not speak English as their first language:

> Whenever I'm involved in a meeting...which involves one of the teams in Germany, I've got a little more sympathy for them because usually I know

---

[7] This is a reference to being a neutral observer, able to see both sides of an issue.

> what they're trying to say, and it's just not getting across to the teams in [Canada]. [9]

Several participants mentioned that the language gap between the two offices often contributed to and was complicated by other tensions or conflicts that existed between them. This is illustrated by the following remark:

> They try to voice their concern as best they can, but...the problem is when they try to communicate at, not a sterner level...but, you know when they're trying to escalate their concern...that's when there's another, there's a bit more of a breakdown in language. They can't be as eloquent or diplomatic about, about their situation. Sometimes people in the Canadian office take offence to what is said...German's can be a lot more to the point. A lot less, I guess, flowery you could say, when they're talking...So decisions can be made in favour of the [Canadian] office just because...people are upset with the [German] office, or they think that they're being unreasonable, which is a bad thing to be marked as when you're working in computers and everybody is supposed to think logically. [9]

Despite the challenges noted by many individuals, several participants also emphasised that they believed that people generally did their best to try and communicate with one another [6,12,13].

4.2.3. Culture

> *I think this is probably a very big part of the problems we had...that people are quite different. [12]*

This category includes anything the respondents referred to as *culture;* it also includes discussions of differences between the two sites that were tied to the organization rather than to individuals. For the most part, the sort of cultural differences that people introduced were related to national or ethnic group membership, although on some occasions, respondents referred to particular ways of doing things within the company, what we might refer to as *office culture* or *organizational culture.*

It is important to be explicit about how I elicited these perceptions of cultural difference. I tried very hard not to assume that difference existed, and did not initiate discussions of difference; nor did I introduce the concept of culture itself. If the person I was meeting with did not mention culture and cultural difference, I did not ask about them. However, if the participant stated that one of things about distributed development that they noticed was the difference between people in the two offices, I asked to them to discuss in greater detail what they meant.

Relevant comments include observations of different ways of doing things or thinking that were identified in the various offices of the company. What is most significant about this theme is the assertion on the part of the participants that significant cultural differences within the organization have existed and continue to exist, and that these differences are strongly tied to particular geographic locations and national identities. Also significant is that, at least in the context of these interviews, people strongly identified themselves and others as being *German* or *Canadian*. The responses that fall under this theme are divided into four sections: administrative and structural differences, different management styles, differences in personal characteristics, and historical influences.

Administrative and Structural Differences

Respondents reported differences at several levels. At one level were administrative and structural characteristics. For example, vacation practices and termination rules vary dramatically between the two countries [9, 18]. Employees in the German office are entitled to six weeks vacation, distributed throughout the year, with at least a three-week summer break [18]. This is not specific to the German office of Chimera: German labour legislation obligates employers to provide six weeks vacation (Bender et al. 2002). In contrast, Canadian labour legislation requires employers to provide only two weeks vacation. Several people mentioned that there is a shorter termination notice period in Canada. A German contributor suggested a link between the short

termination notice periods required in Canada and what he perceived to be a lack of loyalty on the part of Canadian employees:

> On the other side, these people...must always be afraid to be fired, so they never establish such a relationship to the company as we do...If you have no security as employee, why should you give 100% confidence to the employer? It's the same story from both sides. If they give two weeks or ten days [notice], why should you give them more? [18]

If employees in the German office want to work in a second business, during their free time, they are required to make explicit arrangements with their managers. According to one participant from the German office, such extra work is not common there; it is much more common in Canada. He asked how, if you devoted your energy to your main job, you would have extra for a second business or job [18].

Several participants noted that while Germans tend to try and stay with a single organization for most of their career, this rarely occurs in Canada [14, 18]. The notion of *experience* varies between the two offices. According to one participant in the German office, a person who has been working in the company for seven years is considered to be experienced, at least as far as the Germans are concerned. She suggested that in the Canadian office, if you've been working for two years, you are considered experienced within the company [18].

Different Management Styles

Numerous respondents mentioned a difference between German and Canadian management styles [17]. For example, all of the developers, team leads[8] and managers in Germany share offices. However, in the Canadian office, every team lead or manager gets his or her own office [16, 18]. Several participants brought this up, confirming my own experience. Every manager I interviewed in the Canadian office had their own office, while all of the managers I spoke to in Germany shared an office.

---

[8] This is a common term in the software industry for a team leader.

According to several participants, the organizational structure found in the German office is flatter than that found in Canada. Participants noted a clearer hierarchy in the Canadian office with emphasis on moving up the corporate ladder [14,16]. Despite being less organizationally hierarchical, those in the German office follow more formalized protocols for engaging with their colleagues and for addressing those in senior management, or those who have other titles [15]. The following excerpt is from a collection of useful information about the city and the company that I found in my room in the company apartments. It was entitled, "Appendix B: Cultural Norms: Germany":

> Firm handshakes are customary for greetings...and leavings, with the woman or senior person offering first. Appointments and punctuality are essential. Polite forms of speech should be used (eg. Herr or Frau), as well as titles. Third party introductions are also preferred. [21]

These suggestions imply a formality that is not the norm in the Canadian office where most employees address all but the most senior executive by first name only [14].

Disparate levels of formality also extend to the style of meetings conducted in the two offices. In Germany, meetings tend to be structured, somewhat formal, and regulated by an agenda [12,16]. Meetings in the Canadian office tend to be much more open and less controlled. When meetings that include both offices are held, one participant suggested that it is probably true that the Germans adapt to the Canadian style [16].

The following remark summarizes the impact of these different styles: "We had a North American style company with North American style management. We weren't allowed to instil that into [the German office]. They work differently...we promoted the cultural difference" [5]. [9]

---

[9] "Promotion" in this case does not mean the active amplification of or support for difference. Instead, by not actively attempting to bring the offices together, they passively allowed differences to persist.

It was this sort of comment - "they work differently" - that I heard most frequently when people talked about differences between the two offices. Simply put, most people indicated that people from the other office were different, and had different ways of doing things.

People from the German office were variously described as being quiet and reserved, yet opinionated [1]; reluctant to criticize openly; a stubborn group [5]; complicated and overly precise [6]; defensive [14]; and thorough [6]. While explicitly choosing not to use the word *stubborn* to describe the Germans, one participant suggested that they had a tendency to firmly stick to their principles. She described them as relentless, which often made them appear as being not willing to budge on things [7]. A participant from the German office suggested that particular traits were sometimes misconstrued by people from the other location: "One thing which is also maybe a good indicator is...we are very direct, very up front. This may...sometimes be perceived as being insulting for the Canadian guys" [12].

A Canadian participant noted that the Germans wanted to be self-sufficient, that theirs was not a culture of asking for help. She contrasted this with the Canadian office, where people are gregarious and likely to ask for help rather than struggle with a problem and try to figure things out themselves [19].

Canadians are otherwise described as being lax [6], chatty and interested in sharing personal details [12], uncoordinated [12], and prone to indecisiveness [16,15,12]. When issues[10] were sent by e-mail to the Canadian office, those in the German office would complain that they would wait for weeks for a resolution, if they received one at all [6].

---

[10] In this context, *issue* has a very particular meaning. One aspect of the software production process is *issues management*. Whenever an error is identified in the application, and a change or a repair is required, an *issue* that includes the details of the problem is generated and tracked through a formalized issues management system or process.

Highlighting some of these experienced differences, one participant made the following comparison between the individuals in the German and Canadian offices:

> The Germans here like to do the planning and have everything well defined and then start working towards the goal. Where I think the Canadian people are more, talk a lot, without any clear result or clear plan at the end, and then start doing it, and then if it doesn't work, OK, then we discuss it again, and try something else ....My feeling is it's more uncoordinated approach. Not straightforward, like we would like to do it. [16]

Finally, several people talked about the different relationships that employees have with the company. According to one person, the German employees could always be counted on to put the necessary work in to get things done [7]. Although they would gripe and complain, there was always an underlying loyalty. The same could not be said for the Canadian office where getting people to work extra was often like "walking on eggshells". The following remark describes the German employees' ties to the organization: "In our understanding, an [employment] contract is a little bit like a marriage" [18].

Some people indicated that Canadian coders demonstrated a more overt interest in advancing their careers than the Germans did:

> People in [Canada] want to become a manager. Sometime they are much more interested in their career, not just with regards to the work they are doing but more with regards to the title. If they're in the company for some years they want to become a team leader and if [they] ...were a team leader for some time they want to become a manager. [16]

Participants suggested that the Germans, on the other hand, were more interested in a steady job and were much more likely to stay with the firm longer [7,14].

Historical Influences

Finally, some participants suggested that particular cultural traits they believed were shared by many in the German office could be explained by the fact that it was located not just in Germany, but in the former East Germany:

> Maybe another thing that also, maybe is a very special case, you know we have a Canadian company and a German one, but an East German company. So, East Germany has quite a different history. I think that if it had been a relationship between a Canadian and a West German company, things would have been quite different. I don't know in which sense, [better] or worse - could be both. [12]

> It might be that the cultural difference between West Germany and Canada would be much smaller than between East Germany and Canada. It might be that the cultural problem between East Germany and West Germany would be as big as with Canada. I don't know. [18]

Speaking along similar lines, another participant stated that the political and economic context has significant impact in the way the individuals in the German office see themselves, and shapes the way they react to conflict:

> They spent forty years behind the wall, and have had half of their family laid off...This free market economy's kind of put them out into the open, and they tend to be really protective of their jobs to begin with, so they're defensive about everything. If they've made a mistake on it, or they differ with a client, they just argue about it, they're not particularly accepting, 'cause they assume some dire consequence awaits them and they fear being jobless - with a 24% unemployment in the east. [14]

4.2.4. Power

> *And we're starting to... make sure that it isn't somebody in a backroom writing a rule...You get a committee together, and you work on writing your rule. [5]*

Where *time*, *language*, and *culture* emerge from the data as clear categories defined by the participants themselves, *power* is a synthesis category

that contains a range of comments related to decision-making within the organization. This includes a variety of issues, from the relatively straightforward inequality in the distribution of senior management responsibilities between the two offices, to inter-office tension and conflict, to a refusal to observe company standards. It is a broad and imprecise category, but one which captures the myriad of power issues that appear to have been in place within the project and the organization. This theme is divided into two sub-categories: the location of management, and the source of policies and procedures.

Location of Management

Numerous contributors, from both offices, remarked on the inequality between the two offices [7,5,9,12,14,16]; many of these remarks were in reference to the location of management personnel. Chimera headquarters are located in Germany, and the Canadian office is technically an operating division. Nonetheless, during the time period reflected in this study, all upper management relevant to K2 project operations resided in Canada. From the earliest point at which the two offices had a working relationship, things were run out of the Canadian office at the highest level. According to one participant, the company's founders did not plan for the two offices to be unequal; events merely unfolded that way [7].

Recognizing that there was a need to develop management in Germany, the organization took steps to address the management imbalance. One such step was the relocation of a senior manager from the Canadian office to Germany for several years [7,12]. Subsequently, representation of management between the two offices was equalized[11]:

> We have built up more management in the [German] office, of Germans. Whereas before we brought them in, from [Canada]. And ulti[illegible]tely, that had to happen. That's the first issue, is equalize the managem[illegible]

[11] The beneficial effects of this rebalancing did not substantially influence the K2 development team that is the focus of this study.

> structure...And we're ...doing that, and they're starting to get their voice. [5]

Source of Policies and Procedures

According to many participants, the Canadian office controlled most activities around the creation of software processes and standards:

> [The Canadian office] was always the brains. Or at least we felt we were, and we were always calling the shots. So, anytime there was a standard developed, it was written in [Canada] and given to [Germany], and often we didn't ask for their input. And they, most often, had a problem with that. And when it was done the other way occasionally, where [Germany] was starting something, [Canada] had a problem. [5]

Not only was there a common perception that processes were defined unilaterally in Canada, but there was similarly a shared sense that important project decisions emanated from there as well. A Canadian developer described the tension caused by the power imbalance between the two offices:

> A lot of the power, with regards to policy-making and the directions that projects will take, resides in [Canada]. So, there's a lot of friction there. It seems, definitely to the German office, it seems a lot of the time the decisions are always made with a bias towards the [Canadian] office, and without a really open mind towards the ideas that are coming out of satellite offices. [9]

Participants from the German office confirmed this perception:

> From our perspective here, it was actually mostly the feeling that [Canada] wants to make the decisions. They just tell us after the fact how we do things. And they didn't really involve the knowledge of the people here in the decision-making process, or if people were involved, their comments were often disregarded. At least that was the feeling here. [16]

The view that the German office was largely excluded from decision-making and policy-making was not unanimous. First, not all participants talked about power, or mentioned any imbalance in decision-making or policy-making. Second, not everyone agreed that the Canadian office held all the cards, as this excerpt illustrates: "Where was the root of power? Depends who you ask.

[Germany] will tell you it was there. You ask us, we'll tell you it was here. There was no root of power [11]." Another contributor insisted that all perspectives were respected and incorporated into the policies and procedures implemented in the K2 project [8].

Despite these dissenting voices, the majority of respondents who commented on power, or issues that I have classified as *power*, agreed that for most of the company's history, the Canadian office has had substantially more clout that the German office. This discussion is complicated by the blurring of boundaries between experiences with K2 and experiences with other projects in the company. My general sense is that discussions of power in particular reflected a much broader set of personal experiences: people saw the same things happen on multiple projects.

Many participants commented that those in the German office often felt they were not heard. A senior manager described the "antipathies" that existed between the two offices [18]. Some respondents indicated that concentration of control and policy-making in the Canadian office was resented in Germany, while others remarked that this practice led the Germans to ignore the imposed practices altogether.

> I'll tell you one thing that used to go on between these two offices, and...I don't think it's very prevalent anymore...There used to be a time when the people in the [German] office I think were really put off by the people in the [Canadian] office. They had a...feeling that, you know... everybody in [the Canadian office] felt so high and mighty, and that everything was pushed down on them, which was not the case. Sometimes it was the case, but, you know, for good reason. This is where everything started...I think there was some offense taken of that for the longest time, and all the misunderstandings brought about...by that. [1]

Several people who had been involved with K2 and with other projects distributed between the two offices described situations where they or the responsible managers on projects were frustrated by the apparent refusal of people to follow recognized company standards or use existing tools. Some respondents referred to this phenomenon as *N.I.H.*, or *Not Invented Here Syndrome*, common industry parlance to refer to programmers who insist on

creating everything they need, themselves, rather than using perfectly good code created by others.[12] The following excerpt illustrates this point: "And what we witnessed... was "OK, I see this. I know you want me to use it. But I can do it better so I'm going to build another version of it because I can do it better" [5].

Other participants reported being confounded by the persistent resistance on the part of some German developers. One contributor described a recurring situation where members of a team in Germany, having agreed to adopt a particular approach, refused to actually do so: "They would say "yeah yeah, we'll do that" and they would just do what they wanted to do" [4]. After some time, the team would meet again and it would become apparent that they had not done what they had agreed to do. The resulting meeting would often go something like this:

> "Well what do you mean, this issue is coming up again, I thought we had resolved this" "Yes, no we're doing it this way" "But we agreed we were going to do it this other way" "No, no we're doing it this way". So you go through like an hour discussion, and finally at the end, everybody agrees. Two weeks later you come back and they're still--. [4]

A Canadian manager described the resistance it in these terms: "It was that instant thing where, "You're not going to tell me what to do." And for whatever reason, we couldn't stop that from happening" [5]. He suggested that the root of the problem was the relationship between the two offices:

> Probably, it was because, you know, the four or five years before that, it was very dictatorial. I mean, [Canada] called the shots. All the managers were there. We controlled everything, and Germany, you just do what we tell you. And after a while, when they got their confidence and then knew they had good people, that wasn't acceptable to them anymore. And so even if it was a good idea, they resented it. [5]

He further stated that the resentment and resistance developed because those in the Canadian office failed to get the Germans to feel a sense of ownership in the

---

[12] "Not-Invented-Here Syndrome" commonly leads to the practice of "Reinventing-The-Wheel". It can be solved with "Use-Somebody-Elses" or "Swiped-From-The-Best-With-Pride" (http://c2.com/cgi/fullSearch - 18 February, 2003)

processes and decisions created for the project: "In order to get ownership, people have to be involved in deciding how people are going to be doing their jobs, and we just, we didn't understand that soon enough I think" [5].

Another participant suggested that the proximity to decision makers gave the Canadian developers an inherent advantage:

> It's very easy...to enter into a dialog with someone face to face, and also to convince them face to face. So a lot of...decisions are made with the [Canadian] office's concerns...taken into account. And, most of the time [the German office's] concerns then have to be solved via e-mail, which can take two to three days. [9]

4.2.5. Trust, Cooperation and Respect

> *I think the philosophy was, in the beginning...that people would work together, would become a team, work as a team, but this didn't really happen. Because it doesn't happen only... if you have to work on the same product. There's much more to it. [16]*

> *Theoretically we're all working toward the same end, and same product, but you would actually, all joking aside, it became an issue whether or not you would need somebody to mediate between the two sides. [14]*

Like *power*, this category is broad. It includes participants' comments on cooperation, respect and trust, the intangible concepts that are so often associated with the mysterious art of creating successful software development teams. I use the term *trust* to characterize relationships among co-workers. In this study, trust among workers has emerged as more significant than that which describes the relationship between management and workers.

The common report from many participants was that the overall working relationship between the two offices was unhealthy. Respondents reported an overwhelming atmosphere of blame in which problem resolution was often preceded by a prolonged period during which a succession of individuals attempted to cast blame for the problem on the other office [18]. One participant referred to the "subtle, subtle turf wars" [4] that emerged between the Canadian

and German offices. Another made the following observation about the relationship between the two offices:

> Ultimately it was...pretty militant between...the two groups. And I saw it from both sides because I was in both locations. We never, we never really got by that. And we're still working on it, I think. [5]

According to one person [16], this culture of blame developed because people didn't know each other. He explained how, when people know each other, they begin to know that things are not the fault of the other and can achieve an attitude through which they work together. With great distance it's difficult to achieve this attitude. Many participants discussed the challenge of maintaining respect for distant co-workers, as illustrated in the following passage:

> When we meet face to face with people in [Germany] we can get along very, very well and it can be very, very congenial...when we are apart...after a while we begin to think of each other as idiots. And, and it really gets quite venomous...and I'm sure that there are people on both sides of the ocean just shaking their heads and saying "What is wrong with these fools?"...When you cannot communicate the motivation for why things were done the way they were...and that's, that's hard enough to do, you know, when you're co-located...When you can have that kind of problem with somebody even in the same office, just because you didn't talk to them about their motivations, well, in [Germany], between [the German office] and us... there's a lot of friction...And...there's a lot of spite. And, you know I freely admit that I get carried away with it as well, and I know people, people here who do get carried away with it, and...the folks in [Germany] are the same...we know that there are bright people on both sides of the ocean, but invariably, we eventually all think of each other as idiots. [3]

Others confirmed this phenomenon of meeting people, liking them, reaching an understanding with them, returning home, and ultimately reverting to the same negative patterns of behaviour [12,14].

Several people noted that one problem with distributed collaboration is that people tend to interact with one another only when there is a problem [12,14,18]. Thus the contact, already severely restricted, is often exclusively negative: "They don't meet each other. And when they meet it's because the

interface doesn't work. In the end they have...a negative perception of each other [18]." A developer who had spent considerable time in both offices observed the following about how small issues become exacerbated by the tension of distribution, particularly where the practice is undertaken in an atmosphere of suspicion or mistrust:

> If you can't actually walk upstairs, talk to somebody directly, everything becomes really personal. E-mail always sounds cold, or, like you're trying to placate them, or you're patronizing them. They just take it the wrong way, guaranteed. [14]

Further comments reflected the difficulty of establishing and maintaining long-distance relationships with colleagues: "You...lose track of the humanity and the likeable characteristics, and the respect that you have for other people, when the ways in which you can talk to them are very, very limited [3]." Others suggested that separation led to a lack of consideration for the other office, and that it was often easier to just ignore the other perspective [6,7].

There was almost unanimous agreement among individuals from the two offices that the relationship between them was troubled, and that they had a difficult time agreeing on things. The second thing they agreed on was how rapidly barriers were broken down when colleagues had a chance to meet each other and spend time with one another [6]. Spending time in the other office made many participants receptive to other points of view. A Canadian manager recalled her first experience of a conference call meeting in the German office [7]. She described hearing snickering from the other end of the phone line, and realizing how disrespectful the Germans found this to be. The Canadians didn't mean to be disrespectful, it wasn't intentional: they just didn't think about it. From the silence on the phone in Germany, the Canadians assumed that nobody on the other end wanted to participate. The Germans, on the other hand, felt a complete loss of control.

According to one participant, personal contact makes problem solving easier, and the lack of personal contact that is inherent to distributed development creates frustrations: "If there's an issue and you can talk to people

and see they are doing their best, and see they are not lazy, they are really working [16]."

Commenting on her experience with another distributed project, a Canadian participant made the following observation:

> One of the issues for me, what I feel is key in distributed development is essentially relationship building...trying to establish communication. To establish processes, procedures, that's relatively straightforward...but trying to get some of the more human factors things going on, so that people had relationships and respect for each other...was another issue [19]".

4.3. Practices

Practices describe the *goings on* (Orr 1996) of work. At the focus of some of the practices listed here are particular tools, such as e-mail, the telephone, or a configuration management system. Others, such as software process, are amalgams of sophisticated technical solutions and formalized ways of doing things. There are also practices that reflect decisions about how to do things. They are best seen as exacerbating the effects seen in other practices rather than introducing distinct effects of their own. This category includes choices about group size and projects scope. The twelve practices are presented below, in no particular order.

4.3.1. Communicating via E-mail

> *I think the biggest thing with the distributed development thing is that you can't just... run over to somebody's office, get some face-to-face time with them, and ... get to the bottom of it. Because with e-mail you can go back and forth so many times, and still not work out the issue. Where in a phone-call, or just talking over it, you can, you can get past all those misunderstandings that the written language can, can bring about. [1]*

Within the organization, the problems inherent to e-mail communication were widely acknowledged, and people were encouraged not to rely on it. Unfortunately, the eight-hour time zone difference and the language gulf made it the preferred technique for many, especially in the German office. This section

includes observations regarding the specific consequences and challenges encountered when colleagues communicate via e-mail.

The following comment of a senior manager reflects the opinion, shared by several participants, that e-mail is a poor communication medium: "You know e-mail is a bad communication tool. Just for facts, fine, but for feelings and emotions, it's nothing" [18]. A Canadian manager noted how a reliance on e-mail affected consensus building: "It's very hard to build consensus when communication...is low bandwidth. People often don't feel they're heard if all they really can say to you is via e-mail" [19].

Many participants reported that one of the significant challenges posed by e-mail is its vulnerability to misinterpretation:

> I remember when I was, before I went to [Germany], being in [Canada] and having to e-mail people in [Germany]. Responses back that I got, sometimes I was offended at the responses back, or I was upset because I didn't think that I was being taken seriously, things to that effect. A lot of that is just translation, and how their English sounds to us, how our English sounds to them. So, in e-mail I think that was a difficulty. [1]

Yet for many German participants, it was *because* of the language difference between the two offices that they preferred to communicate through e-mail. This was despite being "prodded" by managers to use the telephone.

> Since this was all in English, for the Canadian guys they probably did not really recognize that there was sometimes a problem...The key is at least for myself and for some other people, we prefer really to exchange communication per e-mail. It's much easier for us, and you have time to think about it, and to phrase it correctly. If you are, especially if you are forced to enter an argumentation, to argue, to discuss things, it's - you have another level of complexity. You have to respond to the arguments, or evaluate the arguments, and then to find some counter arguments, and you have to translate it, and you have to translate it well. You feel always something like to be in a handicapped position, since you know how to phrase it in German well so that it sounds appropriate as a response, but you always feel that in English it sounds very plain. [12]

The reluctance on the part of the Germans to telephone the Canadian office was a constant source of frustration for managers. As the following excerpt

illustrates, while many managers acknowledged the language issue, the fundamental inefficiency of e-mail, particularly of problem solving through e-mail, was exasperating:

> Even at the end [after two years of distribution], they would be very reluctant to pick up the telephone and make a call. They'd end up spending two days going back and forth, trying to, basically arguing, bickering about something. And it, because it took twelve hours to get a response, they'd go for two or three days, whereas if they'd picked up the phone they could have resolved it immediately, they could have resolved it. But, because of the language, the Germans, who had English as a second language, often would be very reluctant to pick up the phone. They'd rather form their thoughts in the e-mail. [4]

4.3.2. Communicating via Telephone

> *Sometimes when you're on the phone with somebody, somebody doesn't catch something...if you were seeing them face to face they'd have that blank gaze and would be nodding their head. You know they didn't catch it, they just don't want to say something, maybe they don't want to embarrass themselves in front of people, or they don't think it's important, even though it is. [19]*

Along with e-mail, person-to-person telephone conversation is an obvious tool for members of a dispersed software development team. This section contains the various comments made by participants regarding their experiences with this tool.

The synchronous nature of telephone conversations made them the recommended approach for discussions or problem solving. Although people generally recognized this advantage, the eight-hour time zone difference made it impractical for co-workers to talk to each other regularly as part of their working day:

> For example, what is usually said if there is a problem with an e-mail is "just pick up the phone and talk to the person" to resolve the issue. It's not possible. It takes to the next day...It can't be resolved, and if it's a tough issues you really feel bad the whole night. [12]

According to several individuals, exceptions to this were common. When deadlines loomed, some people worked altered schedules so that they were available to communicate directly with their co-workers in the other office.[13] People who observed atypical work schedules reported that telephone communication in these circumstances was generally effective [19].

Several people placed the various communication techniques available to them on a continuum. At the one end, the barest, least rich form, was e-mail and at the other end, face-to-face conversation. Telephone conversations appeared between the two ends of the continuum. The following comment about the distribution between the company's Canadian and American offices illustrates this:

> I mean between here and [California] it's not so bad because there's only an hour difference, and...it's not a huge problem to at least be able to talk to one another on the telephone...It's interesting, the telephone is a better forum for communicating some of those, not body language but certainly intonation sorts of...or the information that that can carry. But it's a terrible forum for actually discussing technical subjects, because you can't be very precise. [3]

4.3.3. Meeting through Teleconferences

> *The conference calls are often very painful. And you know that not everyone's walking away understanding what was said. [5]*

This subset of telephone communication merits separate consideration simply because of the volume of comments regarding participation in teleconference meetings present in the interview transcripts.

In the early days of the company's distributed collaboration activities, conference calls were conducted using a fairly low quality speakerphone [4]. By the time they became a regular fixture of K2 project activities, the company had invested in specialized teleconferencing equipment.

---

[13] It was people who were involved as something other than a developer who reported doing this [6,8,13]. One manager suggested that programmers were not usually receptive to the notion of modifying their schedules to facilitate interaction with the other office.

According to most participants who talked about conference calls, people in the German office had a very different experience of them than people in the Canadian office. Nobody remarked that teleconferences were challenging for people in the Canadian office.[14] Everyone who brought up conference calls stated that meetings held in this fashion were often demanding and frustrating for people in the German office.

Many participants from the German office expressed a general unease with conference calls: "usually when there were problems, it was said we should set up a conference call, but at least for some of our people that put is in a disadvantaged position" [12]. A manger from that office concurred:

> Or what is also very difficult from our side here, if we have conference calls, and it's in English, to get everything that is discussed on the other side, especially if discussions are going on in the other office, they are talking to each other, then we are sometimes lost and feel it's a waste of time. That actually we could go home...They don't really realize how difficult it is for us. [16]

The following excerpt reiterates the suggestion that this frustration was not always apparent to those in Canada:

> I'll be in a meeting [in the Canadian office] and somebody's going a mile a minute...and then there's silence at the other end of the phone because nobody in [Germany] has understood what they said. And they don't even realize it, and on they go to the next topic, and if they're interrupted and somebody asks them to repeat, they get angry. [5]

Even though teleconferences were a source of frustration for many, one participant argued that they are good to have as they train people to jump into other ways of thinking as they have to adapt to other ways of approaching problems. He argued that this is an essential skillset for a global business [6].

---

[14] I acknowledge the risk of inferring things from what people don't say. Nonetheless, I believe this to be a reasonable inference in this circumstance.

4.3.4. StarTeam®

> *If checking in a file takes...more than seconds, so that you wait just for getting a file locked for yourself...a minute or more, and you are...waiting fifteen minutes for getting files, it's time consuming, it's expensive, and it's bad for the feelings. [18]*

The company used a software application called StarTeam to manage various aspects of the software development process. This application provided a means of managing the sizeable assembly of files and other resources that comprise large-scale projects. It also facilitated the tracking of issues and changes, and the coordination of releases. This section contains comments on peoples' experiences with this system.

Reports on StarTeam were few and mixed. Some observed that the system worked fairly well for developers, and that problems amounted mostly to minor frustrations [4]. Others noted that the data link that many people relied upon when using StarTeam was not fast enough: "We have Star-team servers in every office ...and most of the time people find that servers in remote offices, the bandwidth just isn't there to really use them effectively" [3].

By 'remote servers', this person is referring to the fact that when a team is distributed across offices, they must share the data, or code base. Inevitably, the code base is located in one office, and anyone working on the code base but not located in that office is in a remote office. Where developers are in the same office as the code base, they have a local high-speed connection directly to their data. This is not the case for remote offices, where access to the shared data is achieved through data communication links that are typically not as fast. Consequently, differential quality in access to the code base is common in distributed projects.[15]

®Borland Software Corporation 1994-2003

[15] According to one participant, the tools that handle this data sharing have improved dramatically. It is possible to adopt technical strategies that minimize the experience of differential access. Unfortunately for the K2 team, StarTeam doesn't do this, or at least didn't do it when they were doing distributed development [4].

For much of the period relevant to this study, the main repository for the StarTeam system was located in the Canadian office. People in the German office experienced frequent problems with that system, either because of slow network access or because the system was down [12].

Some participants reported that certain aspects of StarTeam use worked particularly well. For example, at a point in time when the core K2 team was split between Canada and Germany, and when they were in a crunch mode focused on defect correction, the StarTeam system was very effectively used. The work was coordinated mostly through StarTeam, and additionally facilitated by the two key managers on the project who had developed fantastic communication between them. The team was able to fix defects very effectively, mainly because they were primarily fixing defects and the people involved from both locations understood the modules [4]. In addition, because the project was distributed, rules for using the StarTeam system were strictly enforced in a way that would likely not have been the case in a collocated project [4].

4.3.5. Other Tools

This category contains the smattering of comments that people made about various other tools used in the project. One participant mentioned NetMeeting, an application that allows people to share voice, data and images across high-speed networks in real-time. She stated that they have tried to use NetMeeting, but have found that the restricted bandwidth limits the effectiveness [3].

Translation tools were often employed, particularly when people from the German office needed to interpret certain idiomatic expressions or particular phrases used by the Canadians. They would use AltaVista Babelfish or Leo, an online German/English dictionary: "If you used a word that maybe they don't know, immediately they would go to Leo and look it up, and then they would know what you're talking about" [9].

Within the company, on various projects, they used a project management tool called Primavera®. But with the low-capacity network connections between the Canadian office and international offices, it was at best slow and at worst unusable: people in Germany hated it and Malaysians simply didn't use it [4].

When asked if they had used videoconferencing, no one responded affirmatively. Some believed that there was a videoconference-equipped conference room somewhere in the building, but had never used it [3,12].

The shortcoming of many of these techniques is poor useability caused by insufficient network capacity. A senior manager suggested that they should have invested the money to establish high-bandwidth connections between the Canadian and German offices:

> It was going to cost us huge amounts of money to get fat enough pipes to get good performance...from a technical point of view, we should have still probably invested that money because we would have saved it, literally, in peoples' time, but also in frustration. [4]

Finally, a simple but powerful tool was created in order to connect names to people. The K2 team leaders created a site on the company intranet[16] where they posted photographs of everyone working on the project. Their goal was to decrease the personal distance between the teams. It was generally well received, and many participants commented on the positive impact that this had on the team [8,12,15,18].

4.3.6. Software Process

Software Process describes the implementation of formalized rules and procedures to control the way that people create software. Various people described the process needs at Chimera. Processes were needed for three main areas: issues management, configuration management and release management.

®Primavera Systems, Inc. 1995-2003

[16] An *intranet* is a collection of web-pages that can be accessed only within a company's computer network. They are used for the dissemination of information within organizations.

Issues management refers to a consistent method for capturing and addressing defects and changes that emerge throughout the course of the project. Defects were found by testers and logged into their issues management system, StarTeam. Issues went through a definite cycle: *new, open, in progress, fixed* and *verified*. Configuration management refers to the process of version control, including naming versions and managing the multiple versions of the application as a whole, and the individual components. Release management was the process of controlling what features would be included in a release, and what the content of that release would include [8]. Finally, rules for communication and rules about issue response time were put in place. Several people commented on their experiences with software process on K2 and other projects.

One participant was cynical about the company's commitment to software process. She stated that they follow the "copy and bastardize" model of software development and that it is this sort of process that led to the need to create a base version of their main product in the first place [3]. She suggested that part of this could be explained because software developers don't like to follow process: "Communicating with other people, and leaving around and maintaining artifacts that actually describe how the system works...that we tend to ...we tend to give short shrift" [3].

Several individuals observed that in the beginning of K2 distribution, few processes were in place and those that were in place didn't work very well [6,8]. Geographical separation of the team made these processes very fragile. Nonetheless, the situation improved remarkably over the two-year period during which the project was distributed [6].

Eventually, several processes were put in place that helped the project. They began to use StarTeam extensively for issues management. When a tester found a problem, they entered it into StarTeam and those comments were immediately available to everyone, including developers in both offices. In addition, before code was checked into the repository, the developer had to have

that code checked by a second reviewer who had to sign off on the code.[17] This went into the record with the checked in code, and the end result was better quality code that led to fewer frustrations [6,8].

There are multiple views on how processes were created and deployed within Chimera on the K2 project. According to one perspective, processes were largely debated and formalized by the two managers involved in the product. They would do a dry run on the process in the Canadian office, fix the problems, and then give it to the German office. The occasional heated discussion would develop, but then the process would be adopted. It was a quick turnaround [8]. One participant asserted that, even though the German group had input into the processes, there was a perception that this was not the case. She insisted that although the Canadian office "pushed process", ideas came from a lot of people [8].

Others indicated that there was significant resentment of and resistance to the processes devised in Canada and imposed on Germany. Much of this was presented in the *power* category.

Several individuals mentioned that where processes were employed, there was often nobody in place to make certain that they were followed. This problem was not restricted to the K2 project, but was identified in other endeavours within the company, particularly those with a distributed component.

Finally, several people reported that process was not enough to create success in the project. One participant responded that although these rules met with some success, they could not replace "real talking." A person with quite a bit of experience in many different distributed efforts within the organization suggested that the process issues were very manageable. It's always the issues of culture and communication that present the greatest challenges [19].

---

[17] Most systems that manage the multitude of shared files that make up large projects use the terms *check in* and *check out* to describe the way people access files. When a developer wants to work on a particular file, they check it out of the system, also known as the code repository. They can then work on their own local copy of the file. Once the necessary changes have been made, the modified version of the file is then checked back into the repository where it becomes available for other developers to use. This operation is similar to the way in which we check in and check out books at a library.

4.3.7. The Experience Level of Project Members

> *I think that, in a lot of cases, when we started out, everybody grew with the business, so we didn't have anything there. And we didn't even know what was happening from a corporate level. [5]*

Several contributors stated that a lack of experience, on the part of the organization as a whole as well as on the part of individual employees, complicated distributed software development. All of these comments are included in this category, even though the substance of the comments varies.

Some participants talked about experience with distributed development at an organizational level. Although the company had undertaken ongoing distributed collaboration at one level or another for much of it's history, the K2 project represented a substantial escalation in size, scope and complexity. As such, they were entering uncharted waters. They figured out everything as they went along: adoption of new technologies; creation of software processes; implementation of team management techniques; and management of organization growth. These are all things that any growing company must address; only in this case they were exacerbated by the added tension of distribution [4,5,8].

One participant indicated that in the beginning, when the international structure was quite new, it was very much as though the two offices operated completely independently. Now there is a more extensive network of relationships throughout the various offices, and the offices are more integrated [17].

There was also the issue of the technical and functional expertise of the programmers. According to a senior manager, there was a mismatch in the technical experience found on the teams, and this mismatch contributed to the poor relationship that ultimately emerged:

> We had this unhealthy relationship between newcomers and experienced programmers...If you are in conflict with poor programming from somebody who you don't know, it's easy that you map this bad picture to everybody in this group. [18]

Finally, people talked about experience with the *other* culture. One contributor suggested that the cultural differences would have been better negotiated if people in both offices had had experience with the other culture [12]. Others noted that as people had the occasion to meet their co-workers and gain first hand experience with their work habits, abilities and communication styles, the cultural distance seemed to shrink and the working relationships improved [4,6,18].

4.3.8. The Proximity of Team Members

> *'Cause when you're there, you really get to know...you get to know the people a lot better...Even if...they have difficulty speaking English, you learn to understand what they're saying. And that's important, because when you're trying to communicate ideas, it's extremely important to understand who you're talking to. And working with them at that kind of level where you're with them...at least eight hours a day, and talking to them and trying to sort out problems, yeah eventually, it's almost impossible not to come away from a team like that and not have a better understanding of how everybody on the team thinks and how they communicate, so...Yeah definitely being over there, I don't think anyone would think that they hadn't learned something from being over working with people in a different office, in a different part of the world. [9]*

This is perhaps the broadest of the *practice* categories. I used this code for any comments about the impact of proximity on the distributed development process.

The most common observation, one shared by the majority of participants, was that the opportunity to spend time with their colleagues was invaluable. A senior manager stated that whenever they brought people together in one place, "it was like gold" [4]. People indicated that face-to-face meetings served to tear away barriers and foster substantially improved communication.

Team leaders and managers were more likely to travel to the other office than developers [12]. And when people travelled, it was usually to resolve a pressing problem [12, 14]. These problems were not necessarily satisfactorily

resolved as travel was often very targeted. One participant noted the following about travel:

> [It was] an absolute necessity because of such severe conflict between the two, and then when they show up they don't hold hands and make nice after. It's usually, you come up, fight it out, and then somebody's walking away pissed off. [13]

Another participant observed the following about meeting colleagues face-to-face:

> Meeting people...just really tears away, I think, so many barriers, in the whole development, in the whole distributed development effort...I don't think you get that level of comfort dealing with other people until you've actually...met them, and you get...especially between different cultures, you get a sense of...how their language is, and you get a sense of what they're like so you don't misinterpret things. [1]

> Well, and you just, you know, learn what the person's like a lot more, how to deal with them more effectively. You can't do that over the phone. You can't do that over e-mail. You know, you can't take one person who, who may seem very gruff and serious over the phone and in his e-mails, and very abrupt and to the point where it almost sounds rude. You, until you meet them and you say, "Oh, OK, this person is, that's just the way they do business, it's no personal reflection on me, or what the work ethic that I was putting into the project, or anything like that. [1]

Several participants reported that even when a project team was not separated by large distance or multiple time zones, meeting and working together in the same place, even if only for a short time, greatly enhanced the team's effectiveness [3,4]. This enhancement came in the form of the efficiency of direct communication, especially when compared with the inefficiency of e-mail. But there were other, less quantifiable benefits. The following excerpt describes the experience of a developer who travelled to the remote office of a project that was distributed within Canada:

> It gave me much more of a sense of which were the people who where having more problems and which were the people who were having less...it

> was all kinds of things, from...the tone of voice that people had when they asked questions, the expressions on their faces, you know, it's very difficult to tell from an e-mail whether somebody's really having a hard time understanding something, but when you actually, actually see it in their face when they're not quite getting something that you've said...and that, that kind of feedback is really, tremendously valuable. [3]

Many others talked about the subtle nuances of communication that were available when conversing with someone face-to-face but absent when talking on the telephone or exchanging e-mail. Some suggested that it was easier to understand what someone was saying over the phone if you'd had the chance to meet them and get to know them [1,6].

Others suggested that meeting people, putting a face to a name, was a humanizing influence: "If you never meet somebody, except on the phone, you don't really get to know them. I think there has to be that human element that comes out" [5]. A manager from the German office stated that when people had the chance to spend time in the Canadian office, they were more understanding of their colleagues in the other office, acknowledged that they worked hard, and complained less [16].

Despite this universal recognition of the benefits of travel between offices, travel is expensive and was often one of the first things to be cut in times of fiscal restraint [18]. Usually, people travelled only in order to resolve problems, further reinforcing the tensions between the offices as people always met in pressure-filled circumstances [12,14]. Participants remarked that it was usually team leads and managers who spent time in the other offices, rather than the developers who worked together on a regular basis [12,15].

Finally, several participants noted that it's simply much easier to keep on top of a team when you share a workspace [4,18,19]. A manager suggested that even a flight of stairs, let alone a continent or an ocean, introduces an unacceptable wall between team members. Other participants commented on the immediacy of feedback in a collocated project and the frustration created by this lack of immediacy in distributed projects.

4.3.9. The Importance of Group Size

As I have stated elsewhere, Chimera has undertaken distributed collaborative software development in one form or another for most of its history. Numerous study participants who had participated in either K2 or other distributed projects talked about how the size of the teams impacted the process. I coded these comments as *group size*.

Earlier in this chapter, I mentioned the database administration aspect of K2. Participants noted that the distribution of the database administration was, on the whole, successful. They suggested that the small number of individuals who need to coordinate their activities could explain this success, at least in part [11,1]. After all, there were only two DBAs allocated to K2, but thirty or more people in the larger K2 team.

Several participants mentioned their experiences with a project that was dispersed between two Canadian cities. It involved a fairly small team, three to five people, that made the use of contractors working in a remote location who coordinated with developers in the main office. One participant reported that his experience in this project was largely positive [3]. A senior manager found the same project to be equally successful, and reported that this success was likely attributable to there being small team with an exceptionally well-delineated scope.

4.3.10. The Impact of Individual Personalities

> *There are some real people issues. If two people have an antipathy, or one person has a style that's not conducive to distributed development, then that doesn't work. But to me, again that boils down to a management issue. If I know someone has a bad style for distributed development, and I'm responsible for that distributed development, then they should be out of there. [19]*

Several people stated that, in many cases, particular individuals could have a strong impact on the success or failure of a distributed development endeavour. This was most clearly noted by participants in the K2 project. In particular, I

heard many remarks about the two managers who steered the project for much of its duration, one in Canada and one in Germany. A number of people indicated that what success the team had experienced could in large part be attributed to the strong relationship that emerged between these two individuals. They formed an effective conduit of information between the two offices, and did whatever it took for things to work. They developed a friendship as part of the process, and were excellent communicators [6,4,12].

Many people also noted that the success or failure of the distribution sometimes comes down to participants who do not like each other [14]. A senior manager indicated that a successful distributed project needs the right people. If you don't have the right people, with the right skills, the project will ultimately fail [19].

4.3.11. The Style of Individual Managers

The comments included in this category were originally included in the larger theme category, *power*. Upon further reflection, I decided that this subset of what is admittedly all about the exercise of power more accurately represented a practice than a theme. The comments in this section describe specific observable techniques undertaken on a day-to-day basis by managers, techniques that shaped the experiences of the K2 project participants.

One participant repeatedly stated that many of the problems with distribution could be traced to a failure on the part of managers to adequately coordinate, control and make decisions. She indicated that the job of management is to think strategically and make decisions, and that this was not done on the K2 project. Instead, management was "all talk" but made no decisions [15]. Managers should have acted to establish communication links at the developer level, to make them work together as a team, but they failed to do so [15]. Another person echoed this emphasis on the role of the manager, "to make sure that each group is actually talking, that the decisions made are actually executed in that fashion" [19].

One developer stated that in many cases, multiple people were responsible for decisions and the consequence was that it was often difficult to get decisions made [17]. The project was set up so that two individuals were in the position of team leader: one in each office. They were more or less also in the role of project managers, without being officially responsible for that role: "And it's crystal clear that such an unclear responsibility is always bad, independent [of] distributed or constrained development" [18].

4.3.12. *Project Scope*

> *It was massive, and, I think that, yeah I think that's probably part of the reason. It was just sprawling.* [4]

The scope of a project refers to its size and complexity. This category includes the various comments that people made about the impact of project scope on project distribution.

The database administration aspect of the project was presented as an example of a successful distribution. As we have already seen, participants indicated that this success was linked to the small group size and the well-constrained communication needs. These factors contribute to the scope of a project, and participants indicated that the well-defined scope of the database administration side of things was the secret to its success [1,11].

This sharply contrasts the sprawling, unconstrained and enormous scope of the K2 project [15]. A senior manager, when asked how tightly coupled the two offices were, observed that the project was a mess. They had tried to separate things out but it was really tough [7]. Rather than fulfil different roles on the project, the two offices shared everything – hence the tight coupling [8].

4.4. General Reflections on Distributed Development

This chapter closes with an examination of some of the things people said that are highly informative about distributed development, but which defied my

categorization. They include comments about the overall impact of added communication overhead, the economic cost of distribution, why companies might bother to distribute, and some reflections on the characteristics of collocated development.

4.4.1. Overall Communication Overhead

> *Maybe one of these days when the tools that we have to support doing remote communication are that much better, if they ever get that much better, we'll be able to do follow-the-sun[18] without...too much difficulty. But, I don't think there's every going to be much of a substitute for being able to actually hold a conversation with somebody. [3]*

Many people talked about communication in a very broad sense, incorporating language, time zones, culture, as well as one or more practices. It was generally recognized that the success of any programming activity depends on successful communication and doing global work requires a certain overhead to maintain communication channels. According to one participant, when times are tight economically, it is these maintenance activities that are easiest to cut. Sooner or later, however, there are problems in taking this approach. Issues will pop up down the road [6].

Several participants used the term *bandwidth*[19] to describe the character of the communication channel between various teams and offices [3,4,19]. A high-bandwidth link exists between individuals located in the same city, or between cities separated by a minimal number of time zones, whereas the link between the Canadian office and the German office has a very low bandwidth. Bandwidth is further degraded by cultural or linguistic distance. The following

---

[18] *Follow-the-sun* development, proposed as a compelling benefit of distribution, supposedly takes advantage of time-zone separations by having developers in one location, at the end of their workday, hand off their work to developers in another location. These developers do the same thing at the end of their workday. In this way, time zones are exploited, and a 24-hour workday is created. It has yet to be realized.

[19] Bandwidth is originally a technical term that describes signal transmission characteristics of electrical circuits.

excerpt effectively captures the assertion that distributed development inevitably relies upon low-bandwidth communication:

> I guess it all depends on what the real trade-off is...between all the communication barriers and all the problems that arise, that potentially arise...and whatever benefits you may realize... I mean the whole point is you know that the communication bandwidth is bad with distributed development. There's just no two ways around that...there's going to be stuff you just can't communicate that effectively. [3]

4.4.2. The Economic Cost of Distribution

> *If I'm strictly looking at the bottom line - how can I do this most cheaply - I wouldn't choose distributed development. [4]*

Acknowledging the extra overhead, both in terms of communication and management, inherent with distribution, several people remarked that it was simply too expensive. The following excerpt illustrates this type of comment:

> Finally it worked somehow, but if you look at effectiveness and cost versus what was achieved I think it wasn't too successful. There was by far too much overhead, too much problems and...I think it's a lot of wasted money and effort. [12]

4.4.3. Why Bother with Distributed Software Development?

Given the cost, the management headaches, the tensions and the conflicts, why would the company bother with distributed software development? Why would any company bother? Several contributors talked about this.

One participant suggested that it was important to have development shops close to clients [3]. Developers are often the people who provide technical customer support; so distributing development should provide better customer support [18]. Historically, the company began developing versions of its core application in Germany because it was easier to work between Germany and a new European client than it was to develop the application for that client from Canada [4].

Several people argued that if the company wants to sell its product globally, then it must use the services of developers from many parts of the world.

> If the product you're developing has to be distributed internationally, or sold internationally, then the requirements often end up being different in different countries, not just internationalization issues, but look and feel differences, the way people do things. And the distributed development gives you...exposure...to those things right off the mark, whereas you might not discover them until much later when it's actually too late. [4]

A participant from the German office stated that since their clients are global, one way or another the company is going to have to figure out how to build things in an internationally distributed framework [6]. At least as far as clients are concerned, there are always going to be time zone, language and cultural issues. If they can figure out how to address these issues internally, it can only help them service their clients better.

Chimera has adopted a distributed framework for some projects because of corporate mandate: "the president had given the mandate that there would be equal representation in both offices...on the project. So that was just enforced on the project, that we'd been doing distributed development" [4].

Finally, some participants noted that Canadian and American software professionals are expensive [3,19]. They indicated that there was a corporate mandate for distribution because of cost factors, and since there is a growing body of trained developers in locations where salaries are substantially lower, this reason is not likely to disappear.

4.4.4. Problems Also Found on Collocated Projects

> *Even if people sit next to each other, just in the next room or so, they don't always talk. [16]*

Several participants remarked that the problems and challenges they had experienced while working on distributed projects were not all that different from those encountered on collocated projects. One developer observed the following

about her manager: "He certainly has had to grapple with the issues, many of the issues of trying to do distributed development in an organization that arguably doesn't even do localized development all that well" [3].

Many people suggested that distribution is not fundamentally different from collocation in that problems in the former appear in the latter:

> I can give...equally bad examples where people were sitting either on different continents or they were sitting in two adjoining rooms with a door between them...I think that in that case the cultural differences and the time zones increased the threshold. [4]

Other participants reported communication problems within offices, not just between offices [2,15]. Another talked about the tendency not to think of the people in the other office when it comes to problem solving. She said that often 'the other office' was not necessarily in a different country: "Even in the same office, we have the exact same problem. Different floor – out of sight, out of mind" [19].

5. Analysis

In keeping with the ecology metaphor, my first thought was to create an illustration showing how all of the various interdependent parts of Chimera's information ecology fit together. I envisioned a visual representation of the complex web of interdependencies, along the lines of the diagrams found in ecology textbooks (see Figure 1). This biogeochemical diagram shows the interaction among various animal and plant species at the level of chemical reactions. The complex relationships are clearly visible.

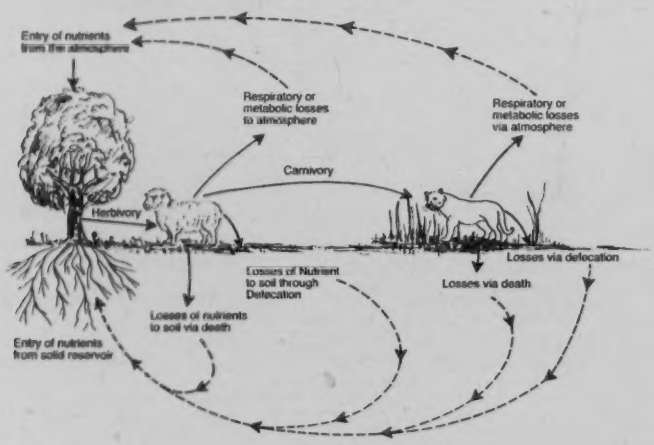
Entry of nutrients from the atmosphere
Respiratory or metabolic losses to atmosphere
Respiratory or metabolic losses via atmosphere
Carnivory
Herbivory
Losses via defecation
Losses of Nutrient to soil through Defecation
Losses via death
Losses of nutrients to soil via death
Entry of nutrients from solid reservoir

Figure 1 - Biogeochemical cycle (from Clapham 1983:53)

Once I tried to do it, I decided not to do it at all. My plan was to represent the various "species" and "processes" present in my data set. In many of the interviews, research participants and I constructed simple diagrams showing where communication links existed and how the two offices could be shown relative to one another. It seemed natural to extend this to the project as a whole. When I sat down to do this, I realized that ecosystem diagrams have no place in them for individuals. In Figure 1 we see the biogeochemical relationship among herbivore, carnivore, tree, air and soil; there is no accommodation for individual sheep or cougars.

I realized that I was unwilling to explore anything such as *programmer-as-species*, with all of the generalizations and simplifications that this would entail. Presumably, more precise classifications could be devised; for example, *fluent-English-speaking-Malaysian-born-working-in-Canada-programmer*. To do this is to reduce people to some essential characteristic, and this seems contrary to the respect for particular circumstances that is so ardently espoused by contemporary anthropology. Even though they are anonymous, the individuals in this study, and their unique experiences, are very important.

Of course at some point, people – including myself - will generalize, mainly because generalizations facilitate learning: the transfer to other circumstances and other situations. This possibility is ultimately where the practical value of such research work lies.

But the whole point of using the ecology metaphor in the first place is to emphasize the fragility, interconnectedness, interdependence, and complexity of the subject. Part of the challenge of distributed work, or any work for that matter, is that it is performed by collections of unique individuals who bring their own experiences, values and ideas into the workplace with them. These individuals cannot be reduced to their most salient characteristics. For the purposes of this study, then, the limits of the metaphor are now visible.

Nonetheless, it remains a powerful framework for understanding the observations described in the previous chapter. As I attempted to devise a way of attacking the substantial body of data, I found it was difficult to separate one concept from another. For example, whenever I began to discuss the particulars of e-mail, I found myself integrating considerations of time, culture, power, language, and trust. E-mail does not serve a single function; nor does it behave the same way in all circumstances. Instead, it is a multifaceted communication tool that must be firmly placed into a network of social relationships, and understood in terms of those relationships. The same can be said of teleconferencing, and software process. This complexity frustrated all attempts at discussing any one practice or theme in isolation from the others.

The information ecology metaphor makes this complexity perfectly acceptable, even expected. We appreciate how fish, bacteria, insects and water

plants are inextricably linked to one another, and that while you can narrow your focus to a study of *Drosophilia*, or carbon cycles, the interdependency is always there in the background. The studies that are narrowly focused on a particular species or process tell us one thing, but they still exist as part of a larger interconnected system.

Keeping this in mind, I decided to isolate a few key systems in the distributed development information ecology: overall communication patterns and constraints; culture; and practices that act as amplifiers.

5.1. Complexities of Communication

While it seems painfully obvious, the quality, content and availability of diverse communication channels are vital factors in the distributed software development context. Herbsleb and Grinter (1999) suggest that many of the problems encountered in distributed projects are also present in most collocated software endeavours. This was the view of many of the study participants: distributed development differs in degree, not in kind, from collocated development. In all but the smallest collocated efforts, success is tied to coordination and control, both of which rely on effective communication among managers, developers and clients. Herbsleb notes that distribution serves to highlight that which is most important, bringing it into sharp relief often because of failure. For example, we can acknowledge the general importance of communication, but in a distributed project, effective communication does not develop on its own. Instead, explicit work must be done to establish and maintain a variety of interpersonal, inter-group, and inter-organizational channels. Those instances where communication breaks down are very informative.

In this project, time, culture, language and trust all articulate with the use of e-mail, the telephone, teleconferences and face-to-face work. Simply put, being separated by an eight-hour time zone difference led people to rely on e-mail. Even though everyone recognized that one-to-one telephone calls were far more effective at resolving problems, for most of the workday, there was nobody at the

other office to talk to when clarification was needed, or problems arose: e-mail was the next choice.

A secondary effect of this temporal separation was the introduction of substantial delay into all aspects of the project. Herbsleb et al. (2000) identify a similar effect in their study of software modification requests. They examined the relationship between the time taken to complete a system modification request, and whether the request involved cross-site work. They point out that any project involves delays, but show that where modifications required input from two sites, completion of the modification request took more than twice the time that it took when only one site was involved. Other research findings support this, suggesting that even a small time-zone separation introduces significant delay (Kobitzsch 2001).

The asynchronous nature of e-mail caused effects linked to the language difference between the two locations. As noted earlier, several German speakers indicated a preference for e-mail communication. They observed that when they compose e-mail, they have time to write, check, and re-read their words before sending them. However, the same language gaps that lead to a preference for e-mail simultaneously left open the door to misunderstanding as words, written to mean one thing, were often read as something totally different. E-mail articulated with culture in a similar fashion. For example, participants reported cases where the Canadians interpreted directness on the part of the Germans, either in e-mail or in teleconferences, as rudeness. Expression is not just linguistically positioned, but also culturally informed.

These effects are also seen in what people said about the telephone and teleconferencing. There was strong agreement among participants that these communication techniques contained significant challenges for those who were not functioning in their native language. One participant described teleconferences as "painful", and several people noted that they usually found it very hard to follow what was going on in the other office, particularly when conversations became animated or argumentative. Both German participants and Canadian participants who had spent significant time in Germany indicated

that many in the Canadian office underestimated the degree to which this was a problem for the Germans.

Another characteristic of the interactions between people in the two offices was that there was a great propensity for misunderstandings and escalated conflicts. Conflicts were blown out of proportion, in part because of the frustrations caused by delayed responses. The participant who described e-mail as an *angrygram* provided one of the best examples of this. Also illustrative are the frequent examples where people described discussions that dissolved into bickering, and the statement by one of the German participants that e-mail is "good for facts, bad for the feelings."

This apparent inflammatory nature of e-mail communication as it was described to me by K2 team members must be understood in terms of certain inherent characteristics of e-mail itself, and its deployment in a brittle relationship of trust between the two locations. The literature that discusses the impact of e-mail and other asynchronous techniques on group dynamics suggests that they have certain innate qualities. Perhaps a better way of thinking about them is in terms of their *affordances* (Norman 1998) or *handles* (Nardi and O'Day 1999). The most significant affordances of e-mail are its asynchronous nature and its detachment from context.

Rocco (1998) found significant implications for the impact of electronically mediated communication on trust building. Specifically, exchange of e-mail messages, or participation in online group discussions alone was not sufficient to foster trust among strangers. Only when participants were given a chance to meet face-to-face, did they then exhibit trust behavior back in the electronically mediated context. Other research echoes these findings. O'Mahony and Barley note that e-mail serves to broaden group participation only in the presence of pre-existing relationships. Sproull and Kiesler (1991) found that, compared with face-to-face communication, people who use e-mail are much less attentive to how others interpret their remarks. Consequently, they are less careful with their words, and are more likely to speak bluntly. They also report that problems are resolved less effectively when communication is restricted to electronic means.

In a study of small globally distributed teams that were culturally diverse, De Sanctis et al. (2001) found a similar linguistic impact to that observed in this study. They looked at the influence of asynchronous communication tools, including group discussions and e-mail, on group formation. The language of the study was English, and those who were not English speakers found that they appreciated the opportunity to carefully compose and examine their remarks before making them publicly available.

An interesting departure from the results of my study is that in the De Sanctis study, the asynchronous tools were credited with the success of the groups. In fact, participants rated group discussion and e-mail as superior to audio conferencing, video conferencing, and instant messaging. This departure can be explained by other differences: most importantly, the composition of groups. Group members consisted of managers from a variety of global companies who came together via the internet in order to learn general management principles. Thus the axis of power is largely absent in De Sanctis' study: there was no pre-existing power imbalance influencing the relationship between the groups. Also, the nature of the trust relationship involving a short-term association of an independent group of highly motivated management professionals is radically different from that observed in this study.

In Chapter 2, I included the reflections of a participant who suggested that the best conditions for software development could be found if you put everyone together in a room. This type of comment was repeated over and over again by people from both locations, leading to a very strong sense that they saw face-to-face interaction as the "gold standard" (Olson and Olson 2000:167).

Overwhelmingly, people stated that it was much easier to overcome language barriers when colleagues were in the same room as one another. They could tell when they were not understood, and figure out ways of making themselves understood. People who did not speak English as their native language found it much easier to follow conversations when the speaker was present.

Also important was the positive impact that fact-to-face meetings had on trust relationships. People noted that face-to-face meetings tore away barriers,

increased comfort levels, and reintroduced the human element. They also talked about the immediate feedback that was available with collocated co-workers. It was possible to find out how work was going, and get a sense of overall time demands. Thus they knew *why* things were delayed or late, not just *that* they were delayed or late

The superiority of face-to-face communication receives considerable support in the literature (eg. Grinter et al. 1999, Heeks et al. 2001, Herbsleb and Grinter 1999, Kavanagh and Kelly 2002, Nardi and Whittaker 2002, O'Mahony and Barley 1999, Olson and Olson 2000). Several authors suggest *common ground* as an entry point to understanding this phenomenon. For example, Kraut et al. make the following observation about common ground:

> Whether communication is face-to-face or remote, speakers can rely on what common ground they have with addressees and side participants when they formulate utterances...To the extent that speakers and listeners share a common work environment and culture, they can draw on this shared knowledge in planning and interpreting utterances (Kraut et al. 2002:148)."

Olson and Olson describe common ground as, "that knowledge that the participants have in common, and they are aware that they have in common" (Olson and Olson 2000:157). Common ground fits in with Nardi and Whittaker's notion of "communication zones", or social fields within which communication takes place (Nardi and Whittaker 2002).

Human interaction has an inherent spatiality: "people and objects are located in space, and their role in an ongoing discussion can be indexed by location" (Olson and Olson 2000:146). When you want to know if you have been understood, you quickly glance at the person you are talking to. If a particular object is the subject of discussion, pointing or nodding toward the object establishes the shared meaning. Linguists refer to the *deictic* references that normally pepper collocated conversations: the pointing references, like "this" and "that", which are very informative but completely reliant on copresence. In this context, it makes sense that when the subject of communication is a visual object, use of a video camera (in the case of a physical thing) or sharing of the

application over a network (in the case of a document or software application) are strategies that have been shown to be promising (Fussel et al. 2000).

Other characteristics of spoken language also factor into the communicative experience. When we can see others talk, we have an extensive armoury of paralinguistic cues that we draw upon (Nardi and Whittaker 2002, Sproull and Kiesler 1991, Olson and Olson 2000). We actively monitor comprehension, we signal our attentiveness, and we get immediate feedback that allows us to repair cases of misunderstanding. These cues are substantially degraded or replaced by other cues when communicators are no longer together.

Conversational participants construct common ground, and they do so continuously (Olson and Olson 2001, Nardi and Whittaker 2002). This occurs not only at the time of the conversation, but also over time, through multiple meetings and interactions. When people spend time with one another, and accumulate shared experiences, these experiences add to common ground, and become available as a communication resource. Nardi and Whittaker suggest we think of the social fields of communication as providing "scaffolding" to the communication process:

> Information exchange is a key goal of communication, but by focusing our theories exclusively on information, we overlook the social processes that scaffold information exchange. (Nardi and Whittaker 2002)

With this in mind, it is not surprising to find that collocated workers find it easier to establish common ground (Olson and Olson 2000, Kraut et al. 2002, Armstrong and Cole 2002). This was certainly reflected in what people who participated in this study had to say. It is also not surprising to find that various tools and techniques function differently in different contexts. For example, a small group of colleagues who speak the same language and know each other relatively well may find a teleconference to be quite effective. However, where there are linguistic barriers, cultural differences, and a poor trust relationship, as was the case in this study, a teleconference can be a frustrating and ineffective communication medium.

The notion of common ground also helps to make sense of why the project web page, containing photos of team members, was so well received. People reported that they appreciated being able to connect a face with a voice and a name, thereby adding a bit of humanity. This simple device seems to have effectively decreased the social distance – providing further *scaffolding* to the communicative space. Carmel and Agarwal (2001) observed a similar effect.

There is a flip side to common ground. When people share considerable common ground, they are able to communicate very effectively with one another, even over the leanest of media (Nardi and Whittaker 2002, Olson and Olson 2000, Kraut et al. 2002). We have all experienced this: think about the last time you talked on the phone, or exchanged e-mails with a close friend. An example of this phenomenon can be found on the K2 project.

Several participants talked about two particular individuals involved in the project, people who arguably held it together; they were the managers, one in Germany and one in Canada. Many noted that these two worked extraordinarily hard to establish a solid communication channel, a good personal relationship and mutual respect. It was accomplished, in part, by spending considerable time in each other's company. One of them reported that, when he travelled to the distant location, he often stayed at the home of the other manager. They also had occasion to talk to each other frequently, thus establishing substantial common ground.

Heeks et al. would call these two managers "bridges" that function "by building strong one-to-one relationships" (Heeks et al. 2001:59) between people at each location. In doing so, they enable highly effective communication and coordination channels to develop between geographically dispersed sites. In this case, their effect on the overall relationship between the two locations may have been more limited than Heeks suggests. Nonetheless, by mobilizing their common ground, the two managers were able to communicate exceptionally well, despite being constrained by the same temporal, linguistic and cultural effects that so seriously limited communication for other project members.

Common ground is a precious resource in distributed software projects: it is like the fresh water or clean air of this information ecology. One of the unifying

threads in the literature is an insistence that effective communication must be supported by opportunities for people to come together and get to know each other, thereby constructing common ground that becomes available in other circumstances.

5.2. Culture

Within the social sciences, particularly in anthropology, the meaning of *culture* has been and continues to be a topic of considerable attention and reflection. Writing in the 1950's, Kroeber and Kluckhon (1952) amassed an impressive collection of definitions and usages of the term. Some emphasized the things that a society produces – material production or material culture – while others stressed the notion of traditions passed down through multiple generations. Still others called attention to the normative aspects of culture – the rules for living in any particular group. The intervening years have not simplified matters; indeed the debate over culture has intensified not subsided.

> In currently fashionable form, culture refers to the knowledge members ...of a given group are thought to more or less share; knowledge of the sort that is said to inform, embed, shape, and account for the routine and not-so-routine activities of the members of the culture. It is necessarily a loose, slippery concept, since it is anything but unchanging. Culture is neither prison nor monolith. Nor of course is it tangible. (Van Maanen 1988:3)

The contingency of culture is captured in Van Maanen's description. The first part – definition of the basic concept – is not particularly problematic. Yes, there are various ways to interpret this definition, and certain aspects of it seem to be vague. These problems can be resolved by a more precise definition, as circumstances require. Instead, contemporary debates pertaining to culture revolve around the two issues suggested in the latter part of Van Maanen's definition: the changeability and polymorphic nature of culture, and the problem of the observer.

Early anthropologists confidently studied "traditional" societies in which they examined, among other things, various structural, linguistic and symbolic systems. They saw these societies as remote, discrete and bounded, by and large unconnected with other societies, particularly the anthropologists' own. Contemporary anthropology rejects the very notion of isolated, discrete cultures, not because of any change in world economy or advanced technologies of travel, but because the notion of discrete cultures was itself questionable as a theoretical concept.

In acknowledging this permeability, anthropologists call attention to the difficulty of drawing boundaries around particular cultural groups and, consequently, particular cultures (Hertzfeld 2001). Contemporary forces of globalisation in which individuals may identify with multiple cultural groups further complicate this boundary-drawing problem. In this atmosphere, then, it is often difficult to speak decisively about *culture*.

This brings us to another important theme in contemporary anthropology: the position of the observer. When early anthropologists studied supposedly discrete cultures, they did so in a way that largely ignored their own role as observers. Culture was seen as something that existed apart from the observer and could be objectively catalogued, described and analyzed. This viewpoint has been largely rejected. Instead, reflexivity is required – observers recognize that the study of culture is an interpretive act, one shaped by both observer and observed (Hertzfeld 2001, Van Maanen 1998).

Given the considerable weight of the culture debate, I introduce it as an analytical category in this study somewhat cautiously. In part, I invoke culture because it is one of the ways that my research consultants positioned their discussion of difference among team members. While there is limited data to support a detailed analysis of what participants meant when they employed the word culture, it is reasonable to infer that they refer to a shared set of ideas, skills and objects that can be attributed to membership in groups defined by region. Culture, in this sense, is a marker of difference. But from this marker of difference we can infer a broader circumstance of group identification, shared ways of thinking, and shared systems of understanding. That there was talk of a

group *over there*, characterized by particular ways of thinking and acting, suggests identification with a group *over here*. I also invoke the concept of culture as it is applied to organizational culture (see discussion in chapter 2). These two interpretations provide the context for the following examination the role of culture in the K2 distributed development project.

Many of the team members I spoke with saw cultural difference as a key factor affecting the distributed development process. As I described in Chapter 4, people frequently categorized their co-workers according to membership in regionally or nationally delineated groups: the Germans were seen to have certain characteristics, and the Canadians were seen to have different characteristics. One participant suggested that it is important to include personnel who are experienced in the "other culture" if a distributed project is to succeed. She suggested that the absence of such experienced people increases the chance of misunderstanding and conflict.

There are many examples in the data to suggest that proximity has a positive impact on the creation of shared cultural understanding. Most participants reported that spending time in the other cultural setting with members of the other group allowed them to come to know the differences in context. They learned how to deal with people better as they came to know them.

The research on cross-cultural organizations is considerable, and the consensus at this time is clear: cultural differences have real and sweeping impact on group dynamics (Nicholson and Sahay 2001, Brannen 1998, DiBella 1996, Tan et al. 1995, and Heeks et al.). Acknowledging cultural difference is not enough, either for the individuals involved or the organizations in which the groups function. Instead, people in culturally diverse groups, particularly those that are geographically distributed, need avenues by which they can come to know the ways of their distant co-workers. They need to synch with them on as many levels as possible, as cultural differences can permeate all dimensions of group work (Heeks et al. 2001).

Restating this idea, Heeks et al. suggest that successful teams – those that are able to synch with each other – often employ people called *straddlers*: "straddlers have one foot in the client's world and one in the developer's world

(Heeks et al. 2001:59)." They are able to mediate between the two cultures, identifying likely areas of conflict. In a sense, they translate a conversation between somewhat distinct cultural groups. Only by actively creating opportunities for these sorts of conversations can groups deal effectively with cultural differences.

The second aspect of culture that is significant to this study is organizational culture. The closest participants came to explicit talk of organizational culture in Chimera was in references to managers and management style. Several talked about how important it is for managers to follow up on processes, making sure that they work and are being followed; they implied that this attention was often lacking at Chimera. One person noted that Chimera provided few incentives for following up on process. Others stated that clearly defined responsibilities and strong decision-making are critical to distributed teams. They suggested that these were not granted adequate priority at Chimera. Several participants stated outright that they were frustrated by the persistent absence of clear decision making. This array of comments suggests that there may be certain factors at the level of organizational culture that affected the K2 project.

The imbalance of power, so much an undercurrent of many discussions, also fits into this category. As far as many of the Canadian managers were concerned, the "intransigence" of many of the developers in the German office was inexplicable. They maintained that they consulted a broad range of interests whenever a new process was developed. Ultimately, however, decisions had to be made and since the personnel responsible for these decisions were often located in Canada, it was frequently the perception that process was driven from the Canadian office. Simultaneously, because people felt that processes and policies were created exclusively in Canada and imposed on Germany, they were frustrated and marginalized. Perhaps this explains the bitter "turf wars" that emerged around issues of process. Both the perceptions that process was imposed, and that it was resisted or ignored, damaged the trust relationship among individuals from the two offices.

5.3. Practices that Amplify

There are two practices that, at first glance, do not appear to be practices at all: group size and project scope. After all, group size and scope are properties of a project, not something people *do*, or *use*. Nonetheless, these categories reflect particular decisions that have lasting implications for how work will be conducted on a daily basis. Someone decides how many people will be allocated to a project, or what will and will not be accomplished as part of the project; these decisions fundamentally shape the project's organization and tasks.

As practices, group size and project scope are particularly interesting. They act like the volume control on a stereo: they impact all of the other practices and themes, and have the capacity to greatly amplify any other effects that are present. With larger teams and more complicated project requirements, there is a greater need for communication. This leaves projects more exposed to problems of linguistic and cultural difference, to conflict related to perceived or experienced power imbalance, and to the exponential delays inherent with large time zone separation.

Most researchers argue that the best projects for distribution are those that consist of relatively independent components or are well defined (eg. Carmel and Agarwal 2001, Ebert and De Neve 2001, and Grinter et al. 1999). Olson and Olson describe the interaction of these factors:

> Tightly coupled work is work that strongly depends on the talents of collections of workers and is nonroutine, even ambiguous. Components of the work are highly interdependent. The work typically requires frequent, complex communication among the group members, with short feedback loops and multiple streams of information. In contrast, loosely coupled work has fewer dependencies or is more routine...In loosely coupled work, there is common ground about the task goal and procedure; it merely needs to be played out. (Olson and Olson 2001:162)

Armstrong and Peter (2002) describe the characteristics of the few dispersed teams that they believe coalesced into *real* teams.

> We have become impressed with the qualities of those few distributed

> work groups we have observed that have become real teams. Such teams are often most effective when modest in size (between six and eight members) and stable over time so that the members get to know each other and establish a track record. (Armstrong and Peter 2002: 187)

The scope of the K2 project was, according to all accounts, both sprawling and laden with ambiguity. They were attempting to amalgamate the features and code from multiple versions of a complex application into a single base version: no simple feat for any project team. While the exact numbers varied over time, the team had as many as 30 members: not huge, but large by many standards. Thus in many ways, even in the absence of temporal, linguistic and cultural factors, it may well have been doomed from the start.

5.4. Ecological Integrity and Information as Commodity

Nardi and Whittaker (2002) propose a more specific formulation of the ecology metaphor – the media ecology – to focus attention specifically on the contexts in which communication technologies are deployed. This, they argue, is the best way to determine the appropriate mix of various technologies for specific distributed work circumstances. They explain that much of the research on communication and communication technology emphasizes their informational aspects. For example, studies of face-to-face communication frequently overlook the ways in which such contact fosters the social relationships that facilitate or scaffold information exchange.

Distributed software development fails so spectacularly in the absence of such scaffolding primarily because the creation of software often relies upon intense collaboration that draw on information strongly tied to the people who create and manipulate it. Communication in this context does not survive the separation into commodity (information) and machinery (specifics of transmission).

Borgmann's distinction between information *about* and *for* reality, and information *as* reality gets a bit tricky in the world of software development. Neither the product of software development (e.g. the application) nor the tools

used in the creation of that product (e.g. compilers or editors) are real, at least in any material sense. Except for the case of embedded systems development, software applications are mostly concerned with the manipulation of symbols, not things. At some point in their use, most applications eventually reengage with the material world. However, that's not what most software developers deal with on a day to day basis. Instead, their reality is an abstract, conceptual one. In my reading of Borgmann, I accept that, despite the abstract nature of their problem domain, software developers regularly deal with information both *for* and *about* reality, although they are also decidedly embedded in information *as* reality.

Developers learn their problem domain by writing, testing and maintaining code. They come to have a history with the code, and learn who else worked on it. They learn the necessary languages, compilers and platforms, and figure out the idiosyncrasies, and the workarounds of these various components. And they often do all of this under the pressure of deadlines. All of this involves an extensive array of knowledge, both explicit and tacit. Some of it is shared formally when, for example, design or architecture documents are created, or comments are added to programs. Much of it is not amenable to codification or formalization. Instead, it becomes the knowledge of individual developers and is only transmitted serendipitously in unplanned conversations that take place over lunch or at the coffee machine, or while troubleshooting a pesky compiler problem. This knowledge contains their information *for* and *about* reality, and supports the practice of software development. The switch to information *as* reality occurs when the richness of this context is forgotten: practice disappears and process is all that remains. An unhealthy information ecology is the result.

5.5. Concluding Remarks

I was simultaneously thrilled and disappointed when, in the late stages of this project, I came upon a paper by James Herbsleb and Rebecca Grinter (1999) titled *Splitting the Organization and Integrating the Code: Conway's Law Revisited.* I was thrilled because their results and my results are strikingly

similar; I was disappointed that I found the paper so long after I had designed and completed my own research.

Herbsleb and Grinter report the results of a research project that studied the integration phase of a project that had been distributed between the German and British offices of a software company. The *integration phase* was the stage at which two parts of a project that had otherwise been quite independent up to that point were pulled together into a single application. They found that there were significant communication barriers caused by splitting the development across two sites. Marked delays emerged, caused by the geographic and temporal separation, even though there was only a one-hour time zone difference. The consequence of these two was a radical reduction in communication frequency and increase in cycle time over what was normally experienced on collocated projects.

Herbsleb and Grinter report a general preference on the part of German developers to use e-mail rather that the telephone, primarily because they were functioning in a second language, and were more comfortable with the extra time to review their words before they had to send them. Conference calls were generally unsatisfactory, especially when more than two people were involved. They found that there were culturally correlated differences in style that impeded communication. The Germans were often very direct while the British invoked more polite styles of address. The result was that they frustrated and irritated each other. Another cultural dimension was a very different outlook on how to handle and work within hierarchies. The British favoured a less status-oriented approach while the Germans were more respectful of hierarchical relationships.

Overall, participants in Herbsleb and Grinter's study indicated that even the smallest changes took much longer when two offices were involved. They also reported a persistent lack of trust and respect, which led to additional delays as well as hard feelings.

The striking similarity between the two studies suggests that there are powerful effects that act across multiple information ecologies. Then again, it might indicate that those who study information ecologies tend to look for and find the same phenomenon. There was, after all, a correspondence in our

methods as well as our findings. What we do know is that the interactions among the people who work in a distributed framework are complex and multifaceted. There appears to be no single factor to explain either the success or the failure of such projects. Instead, the various themes of culture, time, language, power and trust must be acknowledged and actively engaged with in order to support the interdependent practices so important to getting the work done.

6. Conclusion

> *Tools and technology isn't the issue...Anywhere in the world I can have a shared code repository...And I've got internet based tools so I can talk to you. I've got a VPN so I can see your server, I can see what you're working on. If I want to, I can see your machine when you're not there. I can actually run stuff on your machine when you're not there. But that isn't going to solve those problems where you just don't want to do stuff that's helpful to me and I don't want to do stuff that's helpful to you. If we're not a team - that's when the problem starts. [19]*

In the following sections, I conclude my examination of distributed software development by undertaking three tasks. First, I formulate a list of guidelines that I hope will be of some use to practitioners involved in distributed projects. Next, I suggest some promising research questions that have occurred to me as a result of my involvement in this research process. Finally, I situate my project within a broader argument about the role of information in contemporary life.

6.1. Some Guidelines

What follows is a collection of strategies for successful distribution. Some of these have been gleaned from the many first hand reports of experiences with distributed projects that can be found in the literature. Not all distributed endeavors have been outright failures, even though most have been quite challenging. There is much to learn from those who have figured out, in their own circumstances, what practices best support distributed work. The remaining guidelines have been generalized from the preceding analysis.

1) Support the formation of common ground through face-to-face contact

Cultural differences, linguistic barriers and geographic separation will persist as formidable barriers to successful distribution. Common ground equips people to devise effective strategies for addressing these challenges; extra work will always be required in order to establish and maintain common ground.

Travel is the most frequently cited means of creating common ground, although it is also one of the most problematic. Use travel strategically, providing opportunities for people who will need to work together at a distance, to spend some time together in the same location. Make sure that people have a chance to spend time together working productively on shared problem solving and learning, and that their face-to-face meetings do not consist exclusively of tension filled dispute resolution settings in which they are pitted against one another.

If only limited travel is possible, make certain that it is available as a resource at key points in the project lifecycle, particularly at startup (Carmel 1999, Smith and Blanck 2002), which is the most important time for team trust building, and at points of intensive integration. Prudent use of face-to-face time can sustain and enrich subsequent communicative encounters (Nardi and Whittaker 2002). However, the results are not permanent: effective communication zones established through face-to-face contact degrade over time and require regular upkeep (Nardi and Whittaker 2002). If possible, plan travel so that recurring face-to-face meetings acquire a temporal rhythm (Maznevski and Chudoba 2000). Successful distributed teams adapt to this rhythm and become skillful at maximizing their opportunities to spend time with normally distant colleagues.

2) *Choose tools that support the formation of common ground*

One such tool employed in the K2 team was the company intranet site on which photos of team members were posted, thus putting a name to a face. This can be taken further, depending on the interests and inclinations of the users. For example, selected personal information, such as individual milestones, celebrations, interests or hobbies can also help people to learn about one another, thus decreasing social distance.

Desktop/application sharing is another excellent tool for building common ground. If people work together using a shared visual space, their ability to communicate effectively about things occupying that space is improved. When employed in conjunction with a good quality phone or internet voice channel, people are better able to come to a shared understanding about their work.

Finally, internet text chatting is emerging as a powerful tool for creating common ground. Despite being a very simple tool, its spontaneity goes a long way to capturing the incidental water-cooler discussions so critical to collocated collaborations. Because it is text-based, it is attractive to people communicating across linguistic barriers.

3) *Select projects that are appropriate for distribution*

Most practitioners and researchers agree that highly ambiguous and interdependent work should not be distributed. Instead, distributed work should be characterized by well-defined interfaces, clearly specified components and straightforward dependencies (Olson and Olson 2001, Battin et al. 2001). In general, distributed projects that require intensive collaboration should be avoided (Carmel and Agarwal 2001).

4) *Create incentives for long-distance collaboration*

Olson and Olson (2000) suggest that if there is no incentive within an organization for collaborative work, collaboration will not succeed, regardless of whether it is distributed or collocated. They support this argument with research that looked at failure of organizations to adopt collaborative groupware tools. People were expected to start using tools, but were given no real incentive to do so. As a consequence, the tools were strongly resisted until management ultimately dropped them. Incentive structures within organizations must be realigned if people are expected to collaborate with one another across distance.

5) *Talk about 'dispersed' or 'distributed' teams, not 'virtual' teams*

Distributed groups are variously referred to as virtual teams, virtual collaborations, dispersed teams, distributed teams, global teams, and distributed groups. Smith and Planck make the following argument against the term *virtual:*

> The term "virtual" is vague and has become faddish. *Virtual* means almost but not quite. Teams are about performance and dispersed teams fail to perform as well as collocated ones. By calling them virtual, one implies that they are not actual teams, and this is likely to weaken performance expectations...We avoid this term and advise setting up and

> running these teams to make the members' work as real as possible. (Smith and Planck 2002:295)

As language powerfully shapes how we engage with and talk about the world, start your project off on the right foot by acknowledging that your distributed team really exists.

6) Build teams: don't expect them to build themselves

DeMarco and Lister (1987) suggest that successful software development groups undergo a transformation through which they gel, or form a strong and effective team. As far as distributed development is concerned, their outlook is pessimistic: they identify physical separation of team members as a sure way to commit *teamicide*. While this prognosis might be too severe, it does seem clear that distributed teams do not gel spontaneously. It is conceivable that a collocated team might gel simply through familiarity and shared experiences; the probability of this occurring with a geographically dispersed team is small. Strong teams do not form merely because people work on the same project, but must be purposefully fostered, especially when people work and live in different places.

7) Allocate sufficient management resources

It is much harder to manage a distributed team than it is to manage a collocated team. Consequently, managers of distributed projects must be given the requisite resources for more intensive management. They must help the team to explicitly establish communication protocols as well as the team's working approach. Targets, milestones and quality issues must be negotiated at the start of projects (Battin et al. 2001), and mangers must work to keep all team members focused on the task at hand and "on board" (Smith and Blanck 2002:296). Finally, they need to ensure that everyone is doing everything that they said they would do. None of these responsibilities is unique to distributed teams. However, unlike collocated teams, where work often somehow limps along in spite of the manager, distributed work teams must be actively managed right from the start.

8) Employ bridges, straddlers, and liaisons

Distributed teams need to actively employ straddlers, bridges, and liaisons. A straddler is someone who is comfortable or at least familiar with both sites of the organization. He or she is seen as having a foot in each world (Heeks et al. 2001) and serves to act as a translator of sorts between the two sites. A bridge, on the other hand, describes a strong relationship between two individuals, one at each site. Battin et al. (2001) use the term *liaison* to describe a similar relationship, one that was critical to the successful distribution of a large-scale telecommunications project:

> Liaisons were engineers who moved to Arlington Heights for as long as three months. Their responsibility during this period was to meet the Arlington Heights developers, learn the system, help complete the system-level requirements and specifications, and communicate this information back to the development staffs at their home office. (Battin et al. 2001:71)

These liaisons promoted the trust relationships among developers at the various sites, they facilitated communication and they bridged linguistic and cultural barriers – in effect, they constructed and then shared common ground. Bridges, straddlers and liaisons should be sought out and encouraged to create a network of strong ties between geographically dispersed sites.

9) Provide training

Where relevant, provide people with training, especially for new or critical technologies. Do not assume that because they are all technically savvy programmers, they will want to spend valuable programming time figuring out how to make the new internet conferencing tool work. In addition to technical instruction, use training to teach people the appropriate ways of using the tools at their disposal (Smith and Blanck 2002).

10) Use a variety of tools

No single implement is the key to successful distribution. Instead, employ a variety of tools of varying sophistication, combining them for maximum effectiveness (Smith and Blanck 2002)). Use whatever works. Provide reliable

backups for tools that are prone to malfunctioning. For example, make certain that people have access to good old-fashioned telephones in case their internet voice channel starts to act up in the middle of a critical project phase.

11) Acknowledge the value of the keystone species

Software applications are seldom self-explanatory, and computers never fix themselves. Recognize the knowledge held by those who become experts with particular tools, or who are frequently called upon to make systems work. These people are critical to the smooth functioning of complex technological systems – they are the keystone species, and their efforts should be supported and encouraged.

12) Recognize that successful distributed meetings follow the same rules as any successful meeting

All successful meetings are carefully managed; meetings in which the participants are distributed across two or more geographic locations are no exception (Smith and Blanck 2002). Ensure that meetings have a purpose, that expectations are clear, and that an adequate agenda is provided. Provide adequate notification, and establish and enforce clear ground roles. Give those operating in something other than their native language time to interpret the information, and promote their full participation in the meeting.

13) Make it easy for people to figure out who they need to talk to on the other side

At a recent workshop on global software development (GSD Workshop: ICSE 2003, 9 May, Portland, Oregon) the keynote presenter suggested that one reason delays involving cross-site work are larger than those experienced with single site work is that people often don't know who to contact in the other office. By making information about the expertise and responsibilities of the various project stakeholders explicit and easily accessible – supporting the team rather than individuals - organizations should be able to mitigate these delays (Braun and Brügge, GSD Workshop: ICSE 2003, 9 May, Portland, Oregon).

14) *Rotate management across locations*

When managers are rotated across the sites in a distributed project, a general awareness of issues of cultural diversity is established, both for the visiting manager and the local staff (Ebert and De Neve 2001). Experiencing this diversity first-hand helps people to develop effective strategies for recognizing it and coping with it.

15) *Use giant pipes*

Low-bandwidth network connections lead to poor response. Whether that poor response is experienced by someone checking files out of a shared, remote code repository, or by two people in different locations troubleshooting an application together using desktop sharing, frustration is often the end result. Use the highest capacity connections possible. The same logic applies to video-conferencing facilities: better resolution systems with higher refresh rates facilitate better communication, particularly when there is limited common ground between participants.

16) *Never, never, never use e-mail for significant conflict resolution, to make decisions that are risky or ambiguous, or to express anger*

Enough written.

17) *Take a pro-active approach to distribution*

Distributed software development involves certain challenges above and beyond those experienced with collocated projects. Acknowledge these challenges up front and manage them; do not ignore them, hoping that they will eventually resolve themselves.

6.2. On Future Research

There are two directions for future research on distributed development that emerge from this study. First, there is the possibility that organizations

might want to study their own distributed efforts. For that, I propose several things to keep in mind. Second, there are ideas for future research questions within an academic framework, some of which I briefly introduce below.

To organizations considering a study of their own distributed projects, I suggest a careful adherence to some of the principles advocated by workplace anthropology. First, make certain that you really know what people do before devising elaborate plans to improve it: after all, "practice makes process" (Brown and Duguid 2000:91). Remember that detailed knowledge of work practice declines appreciably as we move away from the person who actually does the job. Managers may know what work needs to be done, at an organizational level, but may know considerably less about how work is done. For that, someone needs to get inside the black box of practice (Orr 1996).

Conventional thinking within anthropology suggests that the people likely to notice the most about how things are done are outsiders. Also important is the opportunity for extensive engagement with the subject to be studied. It is often the case that more is learned by being on the ground, watching people work, than by asking them what they do. If possible, then, engage the services of someone outside the group who is able to devote considerable time to the research project. Ideally, that person would have an active role in the project, acting as both participant and observer.

At the workshop on global software development mentioned above (part of ICSE 2003, Portland, Oregon), several discussions on the state of both distributed software development and distributed software development research transpired. It seems that, at this point in time, companies are backing away from distributed projects. Part of this can be linked to the surplus in trained software development personnel that exists because of the downturn in the industry. It is easier to find technical people, so companies are taking advantage of this and consolidating development in fewer locations.

This retrenchment can also be seen as a reaction to the overly enthusiastic way in which many organizations plunged into distribution during the preceding decade. The realization that distribution carries considerable overhead, and that

"follow-the-sun"[1] development is a myth, has caused many organizations to reexamine their distributed projects.

Is there still a case to be made for distributed development research? The consensus of workshop attendees was that the answer to this question is unequivocally "Yes". In the first place, it is likely that the current downturn is temporary, and that shortages in technical personnel will reemerge as a compelling motivator for creating dispersed teams that take advantage of programmers, wherever they might live. Perhaps the more significant reason to continue research is that distributed development is not fundamentally different from collocated development. Distribution strains the development process, highlighting its most fragile parts. Enhanced understanding of these points of stress contributes greatly to the overall knowledge of how software is built.

Two specific directions emerged from my analysis as promising avenues of further research. First, a more detailed consideration of culture and cultural difference, and their impact on distribution, is required. Culture is a complex phenomenon, and is often only superficially addressed in distributed work studies. In much of the literature that I encountered through the research process, the role of cultural factors in much of the work on communications and the creation of common ground is not clear. For example, when Nardi and Whittaker (2002) suggest that one of the benefits of face-to-face communication is that it supports touch, shared activities, and eating and drinking together, how much are these observations tied to particular cultural perspectives? Some researchers suggest that people who work across distance reserve time to share personal details before getting down to business in a teleconference or phone call. Is this ethnocentric, based on North American ideas of acceptable interpersonal relationships, or is it a suitable approach for people from a variety of ethnic and geographic backgrounds? A more extensive consideration of culture seems appropriate.

Second, a more careful investigation of the impact of time-zone separation is called for. Managers in my study had attributed many of their problems to the

---

[1] See description in Chapter 4.

eight-hour time zone separation between the two offices. This seems reasonable, even intuitively obvious, especially when the consequence of time zone differences is little or no workday overlap. However, Herbsleb and Grinter (1999) reported problems arising from as little as one hour of temporal separation. Is any temporal separation too much? Are the same problems reported in this study evident when dispersed teams share a time zone? Further research is required to address these questions.

6.3. Infocentrism

Brown and Duguid (2000) argue convincingly that our contemporary technologically oriented society is fraught with infocentrism: the belief that all things are reducible to information, and any problems are solvable with enough information. In the same way that, to a person with a hammer, everything looks like a nail, they suggest that to a technology designer or information technology manager, everything looks like a problem of not enough information. This is a problem that might be described as an overabundance of information *as* reality.

Software development is, at least at the time that I write this, still a fundamentally embodied activity: *people* make software. Indeed, sophisticated tools and techniques – which are also created by people – facilitate their work, hiding the messy technical details so that developers can operate at powerfully abstract conceptual levels. Nonetheless, people remain at the core of software development. These people are skilled individuals who work collaboratively and interdependently with other skilled individuals in solving complex problems and negotiating vaguely specified and often contradictory and shifting product requirements, all the while dealing quite comfortably with a deluge of esoteric technological information. But information – in the sense of something that can be easily codified, packaged up, and transmitted somewhere else – is not all that they are about.

Rather, they are also people whose engagement with the world, including the workplace, is influenced by past experiences, cultural assumptions, relationships with others, interests, aptitudes, and individual personalities. These

factors are not extraneous to the software development process, but meaningfully inform it. Ultimately, software is created by assemblages of people and machines, existing in intricate systems of relationships and interdependencies that incorporate, but cannot be reduced to, information exchange.

Challenges to infocentrism are not necessarily anti-technology. Rather, by exposing infocentrism we are compelled to critically examine the ways in which we invoke the concept of information. The modern tools with which software developers exercise their craft are indeed impressive. High-capacity networks, fast machines, and sophisticated compilers have dramatically transformed software development, even in the seven years since I worked as a programmer. Telecommunications technologies make it deceptively easy to talk to people on the other side of the planet. All of these systems have phenomenal information-handling capacities, and it is easy to focus on those capacities as the most significant factors affecting distributed software development. But while information matters, and information handling is important, it is people who create, manipulate, and give meaning to that information. Thus in order to appreciate the subtleties of distributed software development, we need to understand how people matter.

Bibliography

Armstrong, David J. and Paul Cole
2002 Managing Distances and Differences in Geographically Distributed Work Groups. *In* Distributed Work, Pamela Hinds and Sara Kiesler, eds. Cambridge, Massachussets: The MIT Press.

Armstrong, David J. and Erika B. Peter
2002 Virtial Proximity, Real Teams. *In* Distributed Work, Pamela Hinds and Sara Kiesler, eds. Cambridge, Massachussets: The MIT Press.

Baba, Marietta L.
1998 The Anthropology of Work in the Fortune 1000: A Critical Retrospective. Anthropology of Work Review XVIII(4).

Battin, Robert D., Ron Crocker, Joe Kreidler, and K. Subramanian
2001 Leveraging Resources in Global Software Development. IEEE Software 18(2).

Borgmann, Albert
1984 Technology and the Character of Contemporary Life: A Philosophical Inquiry. Chicago: University of Chicago Press.

1999 Holding on to Reality. Chicago: The University of Chicago Press.

Brannen, Mary Y.
1998 Negotiated Culture in Binational Contexts: A Model of Culture Change Based on a Japanese/American Organizational Experience. Anthropology of Work Review XVIII(2 & 3).

Brown, John S. and Paul Duguid
2000 The Social Life of Information. Boston: Harvard Business School Press.

Carmel, Erran
1999 Global Software Teams: Collaborating Across Borders and Time Zones. Upper Saddle River, New Jersey: Prentice-Hall.

Carmel, Erran and Ritu Agarwal
2001 Tactical Approaches for Alleviating Distance in global Software Development. IEEE Software 18(2).

Clapham, Wentworth B.
1983 Natural Ecosystems. New York: Macmillan.

Creswell, John W.
1994 Qualitative and Quantitative Approaches. Thousand Oaks, California: SAGE Publications.

DeMarco, Tom and Timothy Lister
1987 Peopleware. New York: Dorset House.

DeSanctis, Gerardine, Matthew Wright, and Lu Jiang
2001 Building a Global Learning Community. Communications of the ACM 44(12).

DiBella, Anthony J.
1996 Culture and planned Change in an International Organization: A Multi-Level Predicament. The International Journal of Organizational Analysis 4(4).

Ebert, Christof and Philip De Neve
2001 Surviving Global Software Development. IEEE Software 18(2).

Franklin, Ursula M.
1999 The Real World of Technology. Toronto: The House of Anansi Press.

Fussell, Susan R., Robert E. Kraut, and Jane Siegel.
2000. Coordination of Communication: Effects of Shared Visual Context on Collaborative Work. *In* Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work, Paul Dourish and Sara Kiesler, eds. New York, The Association for Computing Machinery.

Gluesing, Julia
1998 Building Connections and Balancing Power in Global Teams: Toward a Reconceptualization of Culture as a Composite. Anthropology of Work Review XVIII(2 & 3).

Grintner, Rebecca E., James D. Herbsleb, and Dewayne E. Perry.
1999. The Geography of Coordination: Dealing with Distance in R&D Work. *In* Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, Stephen C. Hayne, ed. New York: The Association for Computing Machinery.

Heeks, Richard, S. Krishna, Brian Nicholson, and Sundeep Sahay
2001 Synching or Sinking: Global Software Outsourcing Relationships. IEEE Software 18(2).

Herbsleb, James D. and Rebecca E. Grinter.
1999. Splitting the Organization and Integrating the Code: Conway's Law Revisited. *In* Proceedings of the 1999 International Conference on Software Engineering. New York: The Association for Computing Machinery.

Herbsleb, James D., Audris Mockus, Thomas A. Finholt, and Rebecca E. Grinter.
2000. Distance, Dependencies, and Delay in a Global Collaboration. *In* Proceedings of the ACM 2000 Conference on Cumputer Supported

Cooperative Work, Paul Dourish and Sara Kiesler, eds. New York: The Association for Computing Machinery.

Herzfeld, Michael
2001 Anthropology: Theoretical Practice in Culture and Society. Oxford: Blackwell Publishers.

Kavanagh, Donncha and Sèamas Kelly
2002 Sensemaking, Safety and Situated Communities in (Con)temporary Networks. Journal of Business Research 55(7).

Kiesler, Sara and Jonathon N. Cummings
2002 What Do We Know about Proximity and Distance in Work Groups? A Legacy of Research. *In* Distributed Work, Pamela Hinds and Sara Kiesler, eds. Cambridge, Massachusetts: The MIT Press.

Kobitzsch, Werner, Dieter Rombach, and Raimund L. Feldmann
2001 Outsourcing in India. IEEE Software 18(2).

Kraut, Robert E., Susan R. Fussell, Susan E. Brennan, and Jane Siegel
2002 Understanding Effects of Proximity on Collaboration: Implications for Technologies to Support Remote Collaborative Work. *In* Distributed Work, Pamela Hinds and Sara Kiesler, eds. Cambridge, Massachussets: The MIT Press.

Kroeber, A. L. and Clyde Kluckhohn
1952 Culture: A Critical Review of Concepts and Definitions . Cambridge, Massachusetts: Peabody Museum.

Marshall, Catherine and Gretchen B. Rossman
1995 Designing Qualitative Research. Thousand Oaks, California: SAGE Publications.

Maybury, Mark
2001 Collaborative Virtual Environments for Analysis and Decision Support. Communications of the ACM 44(12).

Maznevski, Martha L. C. K. M.
2000 Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness. Organization Science 11(5).

Nardi, Bonnie A. and Vicki L. O'Day
1999 Information Ecologies: Using Technology with Heart. Cambridge, Massachusetts: The MIT Press.

Nardi, Bonnie A. and Steve Whittaker
2002 The Place of Face-to-Face Communication in Distributed Work. *In* Distributed Work, Pamela Hinds and Sara Kiesler, eds. Cambridge, Massacusetts: The MIT Press.

Newell, Sue, Shan L. Pan, Robert D. Galliers, and Jimmy C. Huang
2001 The Myth of the Boundaryless Organization. Communications of the ACM 44(12).

Ngwenyama, Ojelanki K.
1998 Groupware, Social Action and Organizational Emergence: On the Process Dynamics of Computer Mediated Distributed Work. Accounting Management and Information Technologies 8.

Nicholson, Brian and Sundeep Sahay
2001 Some Political and Cultural Issues in the Globalisation of Software Development: Case Experience from Britain and India. Information and Organization 11.

Norman, Donald A.
1999 The Invisible Computer. Cambridge, Massachusetts: The MIT Press.

O'Mahony, Siobhan and Stephen R. Barley
1999 Do Digital Telecommunications Affet Work and Organization? The State of Our Knowledge. Research in Organizational Behavior 21.

Olson, Gary M. and Judith S. Olson
2000 Distance Matters. Human-Computer Interaction 15.

Orr, Julian E.
1996 Talking about Machines: An Ethnography of a Modern Job . Ithaca, New York: ILR Press.

Pacey, Arnold
1983 The Culture of Technology. Cambridge, Massachusetts: The MIT Press.

Pfaffenberger, Bryan
1992 Social Anthropology of Technology. Annual Review of Anthropology 21.

Postman, Neil
1992 Technopoly: The Surrender of Culture to Technology. New York: Knopf.

Reeves-Ellington, Richard
1995 Organizing for Global Effectiveness: Ethnicity and Organizations. Human Organization 54(3).

Reeves-Ellington, Richard
1998 A Mix of Cultures, Values and People: An Organizational Case Study. Human Organization 57(1).

Reeves-Ellington, Richard H. and Adele Anderson
1997 The Ethnology of Information: Cultural Learning through Cooperative Action Research in a Multinational Firm. Accounting, Management and Information Technolgy 7(3).

Rocco, Elena.
1998. Trust Breaks Down in Electronic Contexts but can be Repaired by Some Initial Face-to-face Contact. *In* CHI '98 Conference Proceedings, Claire-Marie Karat and Arnold Lund, eds. New York: The Association for Computing Machinery.

Smircich, Linda
1985 Is the Concept of Culture a Paradigm for Understanding Organizations and Ourselves. *In* Organizational Culture, Peter J. Frost, Larry F. Moore, Maryl R. Louis, Craig C. Lundberg, and Joanne Martin, eds. Beverly Hills, California: Sage Publications.

Sproull, Lee and Sara Kiesler
1991 Connections: New Ways of Working in the Networked Organization. Cambridge, Massachusetts: The MIT Press.

Suchman, Lucy
1995 Making Work Visible. Communications of the ACM 38(9).

1999 Working Relations of Technology Production and Use. *In* The Social Shaping of Technology, Donald MacKenzie and Judy Jajcman, eds. Buckingham, Great Britain: Redwood Books.

Tan, B. C. Y., R. T. Watson, and Wei Kwok-Kee
1995 National Culture and Group Support Systems: Filtering Communication to Dampen Power Differentials. European Journal of Information Systems 4.

Van Maanen, John
1988 Tales of the Field. Chicago: The University of Chicago Press.

Van Maanen, John and Stephen R. Barley
1985 Cultural Organization: Fragments of a Theory. *In* Organizational Culture, Peter J. Frost, Larry F. Moore, Maryl R. Louis, Craig C. Lundbert, and Joanne Martin, eds. Beverly Hills, California: Sage Publications.

Winner, Langdon
1986 The Whale and the Reactor. Chicago: The University of Chicago Press.

Appendix A

Sample Letters of Introduction

1. Sample of letter of introduction sent to potential German participants

Hi XXX,

My name is Lori Kiel, and I am a graduate student at the University of Alberta in Edmonton. I have been given your name by XXX and XXX, with the consent of XXX[1]. I will be in Germany between May 13 and May 22, and would like to meet with you and interview you as part of a research project that I am conducting on distributed software development at Chimera.

The research project involves interviews with people who participated in the distributed development of K2 between October of 1999 and May of 2001. I have already talked to some people in the Edmonton office. Because the study will be more complete with contributions from participants in German, XXX and XXX have provided me with a list of people in your office who I should speak to and your name was on this list.

Your participation in this study is voluntary and your contributions will remain anonymous. In any results that I report, the name and identity of participants will not be provided. You can change your mind about your participation at any point in time. If you choose to withdraw from the study, I will respect your wish to do so.

As I will only be in Germany for a limited time, I would like to make as many arrangements as possible through e-mail. If you choose not to participate, please advise me of this and I can exclude you from the study. If you are interested in proceeding, I will contact you in the next few days through e-mail in order to schedule an interview.

Thank you for your time. I look forward to meeting you. If you have any questions about either the research project or the interview process, please do not hesitate to ask.

Regards,
Lori Kiel
University of Alberta
Edmonton, Canada

---

[1] The first two names identify the manager of the K2 project and the senior manager who had facilitated my continued research involvement at Chimera. The third name identifies the division manager.

2. Sample of letter of introduction sent to potential Canadian participants

Hi XXX;

My name is Lori Kiel, and I am a graduate student at the University of Alberta. I have been given your name by XXX and XXX, with the consent of XXX. I would like to meet with you as part of a research project that I am conducting on distributed software development at Chimera.

The research project involves interviews with people who participated in the distributed development of K2 between October of 1999 and May of 2001. The interview will take approximately one to one and a half hours. Your participation is voluntary and your contributions will remain anonymous. In any results that I report, the name and identity of participants will not be provided. You can change your mind about your participation at any point in time; if you initially agree to participate but later choose to withdraw from the study, I will respect your wish to do so.

Providing you are interested in participating in the study, please let me know right away and we can arrange a time and place to meet. Should you prefer to speak directly to me about the research project, the interview process, or in order to schedule the interview, please feel free to call me at NNN-NNNN.

Thanks very much for your time.

Regards,
Lori Kiel
University of Alberta
Edmonton, Canada

Appendix B

Sample Interview Consent Form

Interviewer: Lori Kiel
Department of Anthropology
University of Alberta
Edmonton, Alberta, Canada
T6G 2H4

Interviewee:_____ Date:_____

As the Interviewee, I have been fully informed of the following points before proceeding with the interview:

1. My participation in this research is completely voluntary and I understand the intent and purpose of this research.
2. I understand that if I choose not to participate, my supervisor will not be informed of this.
3. Upon my request, I understand that my identity will be kept confidential and that I have the right to withdraw from this research at any time.
4. I know that I may refuse to answer any questions and that I may withdraw at a later data. Any information provided by me can be destroyed at any time upon my request.
5. I am aware that others will be reading the results of this research and that this research will eventually be published. In any publication, I will not be identified by name.
6. I know that if I have been a participant in an interview, and the researcher chooses to use a direct excerpt of a transcript of that interview in a publication, I will have the opportunity to verify the content of that excerpt prior to publication.
7. Additional conditions for my participation in this research are noted here:

_____

_____

_____

I will receive a copy of this contract.

Signatures: Interviewee_____

Interviewer_____

Appendix C

Sample Interview Questions

For many of the interviews, especially the earlier sessions, I created a list of questions that I intended to use as a general guide for the conversations. My hope was that they would provide a common structure for the interviews. Ultimately, this never really transpired. In part, this is because I found it a challenge to simultaneously ask meaningful questions, listen carefully to what people were telling me, take useful notes, process what was being said, and plot the future course of the interview by sticking to a list of questions. The most useful strategy was to carefully attend to the conversation and rely on my own immersion in the topic for structure. Ultimately, most of the more interesting conversational turns resulted from something said in the interview, something that probably could not have been anticipated: it was important to follow those threads. That being said, the following collection of questions indicates something about what I was looking for, at least at the beginning of the project.

- What is K2?
- When was its development distributed?
- How was the work divided between the two offices?
- How much had the people in the two offices worked together before K2 distribution?
- What was the role of the Canadian office?
- What was the role of the German office?
- How many people were on the K2 project team?
- What sort of work do you do at Chimera?
- How long have you been with the company?
- What was your role on the K2 team?
- To what extent did you work, on a day-to-day basis, with people in the other office?
- What type of work was shared between the offices?
- What channels did you use to communicate with the people in the other office?
- Have you spent time in the German/Canadian office?
- Why did you travel there?
- How does the product development cycle work?
- Did you enjoy working on the K2 project?
- How did your experiences on the K2 project differ from those on other project teams?
- As someone with direct experience, what do you think about distributed software development?

Appendix D

Ethics Statement and Approval

MA Research Proposal - Lori Kiel (Department of Anthropology)

This research project will be conducted in accordance with the research ethics guidelines provided by the Department of Anthropology at the University of Alberta. According to those guidelines, known as the Tri-council Ethics Guidelines, a set of eight principles must be observed in any project that involves research on human subjects. All of my research activities will be founded on a fundamental respect for and adherence to these principles. Each principle is addressed in detail below.

Principle 1: Respect for Human Dignity

The ethics guidelines require that the "multiple and interdependent interests of the person - from bodily to psychological to cultural integrity" be protected in the course of completing research. Due to the nature of my research project, I do not foresee any circumstances under which my presence at Chimera[1] will in any way threaten the bodily integrity of my research consultants. By presenting my research methods and goals openly to all potential consultants, and by adhering to the principles of free and informed consent and of confidentiality discussed below, I will treat all participants with respect and dignity, thereby protecting their psychological and cultural integrity.

Principle 2: Respect for Free and Informed Consent

According to the ethics guidelines, ensuring respect for free and informed consent, translates, in principle, "into the dialogue, process, rights, duties and requirements for free and informed consent by the research participant". At a minimum, all Chimera employees involved in this project will be explicitly informed of my status as a researcher, the objective of my research, the methods I propose to use, and how I plan to use my research data (ie academic publications, internal Chimera reports). They will be informed that they may choose not to participate, and that at any time during my research activities, they may choose to terminate their involvement. There are potential complications to this principle in a business setting. Are subjects who are told by their bosses that they should make themselves available to me as a researcher really consenting? In order to address this concern, I will explicitly state to all research participants that they can choose not to participate, and that the status of their participation will not be communicated to either their superiors or to other employees. In interview sessions, I will present interviewees with a consent form that outlines all of the above information.

Principle 3: Respect for Vulnerable Persons

---

[1] In my original ethics application, the company name and project group were included. These have been removed here to retain participant anonymity

Vulnerable persons, according to the research ethics guidelines, are "those whose lack of competence or decision-making capacity renders them vulnerable", citing children and institutionalized persons as examples. No individuals that fall within the guidelines' description of vulnerable persons will be involved as participants in this research project.

Principle 4: Respect for Privacy and Confidentiality

The research guidelines specify that all research on human subjects must satisfy conditions regarding respect for privacy and confidentiality. The researcher must "protect the access, control and dissemination of personal information". Anonymity must be guaranteed where desired and appropriate. The concepts of privacy, confidentiality and anonymity exist at multiple levels in this research project. At one level are the employees of Chimera, those whose activities will be observed and recorded, and whose opinions, ideas and beliefs will be elicited during interviews. As researcher, my ability to understand the technology of distributed development at Chimera will depend, at least in part, on my ability to gain the trust of those who do distributed development. That trust rests significantly on the guarantee that personal information revealed to me in my privileged position as researcher will remain private and confidential.

At another level is the corporation itself. Chimera will be guaranteed anonymity, if that is what they desire, in all academic publications that emerge from this research project. Any interim internal reports generated specifically for Chimera will remain the property of Chimera. Chimera may require that all researchers involved in this project sign a legal confidentiality/trade secret agreement in order to protect their own interests. Such an agreement will be honoured.

Principle 5: Respect for Justice and Inclusiveness

According to the guidelines, "no segment of the population should be unfairly burdened with the harms of research". Duties are also imposed "neither to neglect nor discriminate against individuals or groups who may benefit from advances in research". The population in this case consists of the Chimera software developers. With the exception of concerns regarding the use of my findings (to be discussed under Principles 6-8), I do not foresee any harms of this research and, as such, do not anticipate that either harms or benefits will be unfairly apportioned.

Principle 6: Balancing Harms and Benefits *and*
Principle 7: Minimizing Harm *and*
Principle 8: Maximizing Benefit

The final three principles to be considered specify that the researcher must balance the harms and benefits of the research, that harm to the research participants must be minimized, and that benefit to the participants, to other individuals or to society as a whole must be maximized. According to this principle, "foreseeable harms should not outweigh anticipated benefits". In interpreting and addressing this principle, perspective is of paramount importance when it comes to determining what constitutes a harm and what

constitutes a benefit. Baba (1998) argues that much of what the anthropologist in the work setting is engaged in is a process of mining the knowledge of the workers and presenting that knowledge to management. The benefits consist of the possibility for an improvement in the work process, in productivity, and, potentially, in the overall profitability of the organization, things with the potential to benefit all employees. However, there are no guarantees that management will be progressive and will use the newfound knowledge to improve the workplace. Baba suggests that the way to address this problem is to openly discuss it with employers, negotiating the circumstances under which the knowledge that working people volunteer to the anthropologist is made explicit and available to their employers.

In addition, Suchman argues that researchers need to be aware that, in the process of studying workplaces, they are "unearthing the underlying work processes" (1995: 56) and recording these processes in some sort of representational form. She asserts that all representations are tied to "perspectives and interests" (Suchman 1995:59): all representations require context. According to Suchman, "critical ethnography rejects the notion that we can somehow innocently write descriptions of others, whether in the service of understanding or of intervention" (Suchman 1995:61). The ethnographer does not serve to objectively record, but instead interprets observations in a manner that necessarily reflects upon their own positioning. This is not a fatal flaw in ethnographic research but is instead "a fundamental aspect of representational work, to be understood and incorporated into our practices and into what we produce" (Suchman 1995:63).

Such questions are endemic to projects that involve anthropology in workplace settings and, arguably, extend to any anthropological research with a potentially applied component. Nonetheless, every effort will be made to ensure the protection of the research participants at whom the guidelines are directed: the software developers and managers working on the K2 distributed development project. And any representation of this fieldwork will incorporate the principles of openness and reflexivity that characterize and are expected of all modern ethnographic work.

I have read, understood and will adhere to, the guidelines of the Tri-Council Policy Statement-Ethical Conduct for Research Involving Humans, and the "University of Alberta Standards for the Protection of Human Research Participants" (GFC Policy Manual, Section 66) and agree to abide by these standards in conducting research.

UNIVERSITY OF ALBERTA
CERTIFICATION OF ETHICAL ACCEPTABILITY

DEPARTMENT OF ANTHROPOLOGY
HUMAN RESEARCH ETHICS ADVISORY GROUP

APPLICANT'S NAME: Lori KIEL

APPLICANT'S DEPARTMENT: Anthropology

APPLICATION TITLE: MA Thesis Proposal

DATE SUBMITTED: Feb 2002

The application noted above was reviewed by the Department of Anthropology Ethics Advisory Group. The Advisory Group was constituted and the decision was rendered as specified in the "Tri-Council Policy Statement – Ethical Conduct for Research Involving Humans," and the "University of Alberta Standards for the Protection of Human Research Participants", 1999 GFC Policy Manual, Section 66. The committee reviewers for this application are listed below.

This is to certify that the project and/or procedures outlined in the application were found to be acceptable on ethical grounds and to be generally in accord with policy guidelines as laid down by this University for such research involving human participants.

FEB 22 2002

Dr. Cliff Hickey, Professo.
Department of Anthropology Ethics Advisory Group
Statutory Member of Faculty of Arts Research Ethics Board

Date Approved